US009448815B2

(12) **United States Patent**
Pahlavan et al.

(10) **Patent No.: US 9,448,815 B2**
(45) **Date of Patent: *Sep. 20, 2016**

(54) **SERVER-SIDE COMPUTING FROM A REMOTE CLIENT DEVICE**

(75) Inventors: **Babak Pahlavan**, Palo Alto, CA (US); **Daniel Ernesto Barreto**, San Francisco, CA (US); **Curtis Schwebke**, Los Gatos, CA (US)

(73) Assignee: **WYSE TECHNOLOGY L.L.C.**, San Jose, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 831 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **12/546,663**

(22) Filed: **Aug. 24, 2009**

(65) **Prior Publication Data**

US 2010/0269046 A1 Oct. 21, 2010

Related U.S. Application Data

(60) Provisional application No. 61/169,664, filed on Apr. 15, 2009.

(51) **Int. Cl.**
G06F 3/048 (2013.01)
G06F 9/44 (2006.01)
(Continued)

(52) **U.S. Cl.**
CPC ***G06F 9/4445*** (2013.01); ***G06F 3/1462*** (2013.01); ***G09G 5/003*** (2013.01); *G09G 5/42* (2013.01); *G09G 2340/045* (2013.01); *G09G 2360/18* (2013.01); *G09G 2370/027* (2013.01)

(58) **Field of Classification Search**
CPC G06F 9/4445; G06F 3/0484; G06F 3/048; G06F 3/0481
USPC 715/751, 740
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,777,880 A 10/1988 Beattie et al.
5,220,657 A * 6/1993 Bly G06F 17/30171 707/E17.007

(Continued)

FOREIGN PATENT DOCUMENTS

EP 1998245 3/2008 ..... G06F 3/048
WO WO 95/35535 12/1995

(Continued)

OTHER PUBLICATIONS

Sanz, A., et al., "XMLBased Integration of Web, Mobile and Desktop Components in a Service Oriented Architecture", Aug. 29, 2008, International Symposium on Distributed Computing and Artificial Intelligence 2008, vol. 50/2009, pp. 565-573.

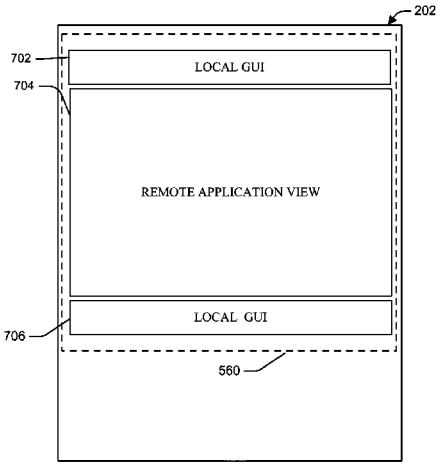
(Continued)

Primary Examiner — Ryan Pitaro
(74) *Attorney, Agent, or Firm* — Baker Botts L.L.P.

(57) **ABSTRACT**

Examples of systems and methods are provided for server-side computing from a remote client device. A system may allow rendering at a local client side a composite view including a local graphical user interface and a remote application view associated with a remote application running at a remote server. The system may comprise a local view module configured to generate a local graphical user interface (GUI) for controlling the remote application remotely from the local client side, configured to receive one or more user inputs to the local GUI, configured to determine a function corresponding to the one or more user inputs, configured to generate a command message based on the determined function, and configured to direct the command message to the remote server. A composite view comprising the local GUI and the remote application view may be provided to a local display at the client device.

21 Claims, 17 Drawing Sheets



(51) **Int. Cl.**

G06F 3/14	(2006.01)
G09G 5/00	(2006.01)
G09G 5/42	(2006.01)

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,581,686	A	12/1996	Koppolu et al.	715/784
5,909,545	A *	6/1999	Frese, II	G06F 9/4445 709/208
5,918,039	A	6/1999	Buswell et al.	
5,961,582	A	10/1999	Gaines	
6,020,881	A	2/2000	Naughton et al.	
6,085,247	A	7/2000	Parsons, Jr. et al.	
6,128,010	A	10/2000	Baxter et al.	
6,286,003	B1	9/2001	Muta	
6,341,097	B1	1/2002	Hsu et al.	
6,430,609	B1 *	8/2002	Dewhurst et al.	709/220
6,483,813	B1	11/2002	Blencowe	
6,510,229	B1	1/2003	Geile	
6,518,965	B2	2/2003	Dye et al.	
6,631,425	B1	10/2003	Helland et al.	719/332
6,654,784	B1	11/2003	Wei	
6,710,790	B1	3/2004	Fagioli	
6,784,855	B2	8/2004	Matthews et al.	
6,836,885	B1	12/2004	Buswell et al.	
6,920,502	B2	7/2005	Araujo et al.	
6,938,221	B2	8/2005	Nguyen	
6,950,991	B2 *	9/2005	Bloomfield et al.	715/738
6,990,477	B2	1/2006	Cotner et al.	
7,093,003	B2	8/2006	Yuh et al.	
7,177,902	B2	2/2007	Hubbard	
7,213,228	B2	5/2007	Putterman	
7,242,406	B2	7/2007	Robotham et al.	
7,274,368	B1	9/2007	Keslin	
7,293,243	B1	11/2007	Ben-Shachar et al.	
7,430,681	B1	9/2008	Hobbs	
7,472,075	B2 *	12/2008	Odinak	G06Q 30/0266 455/150.1
7,475,421	B2	1/2009	Abdo et al.	
7,489,306	B2	2/2009	Kolmykov-Zotov et al.	
7,493,560	B1	2/2009	Kipnes et al.	
7,502,754	B2	3/2009	Campbell et al.	
7,512,906	B1	3/2009	Baier et al.	
7,562,297	B2	7/2009	Carroll et al.	
7,577,924	B2	8/2009	Nguyen	
7,584,505	B2	9/2009	Mondri et al.	
7,607,128	B2	10/2009	Arthurs et al.	
7,685,539	B2	3/2010	Nguyen	
7,703,047	B2	4/2010	Keely, Jr. et al.	
7,705,829	B1	4/2010	Plotnikov	
7,711,366	B1	5/2010	O'Neil et al.	
7,747,086	B1	6/2010	Hobbs et al.	
7,870,193	B2	1/2011	Hintermeister et al.	
7,870,496	B1	1/2011	Sherwani	
7,895,521	B2	2/2011	Bhogal et al.	
7,996,461	B1	8/2011	Kobres et al.	
8,024,407	B2	9/2011	Harwood et al.	
8,078,164	B2 *	12/2011	Ganesan	455/432.1
8,996,658	B2	3/2015	Anbuselvan	
2001/0056422	A1	12/2001	Benedict	
2002/0047863	A1	4/2002	Hyman	345/744
2002/0057295	A1	5/2002	Panasyuk et al.	
2002/0075160	A1	6/2002	Racz	340/825.69
2002/0083098	A1	6/2002	Nakamura	
2002/0118175	A1	8/2002	Liebenow et al.	
2002/0126099	A1	9/2002	Engholm	
2002/0126144	A1	9/2002	Chenede	
2002/0129096	A1	9/2002	Mansour et al.	709/203
2002/0165993	A1	11/2002	Kramer	
2003/0087601	A1	5/2003	Agam et al.	
2003/0115296	A1	6/2003	Jantz	
2003/0122856	A1	7/2003	Hubbard	
2003/0160813	A1	8/2003	Raju	
2003/0177322	A1	9/2003	Crockett et al.	
2003/0212770	A1	11/2003	Kotnur et al.	709/220
2004/0141010	A1	7/2004	Fitzmaurice et al.	
2004/0205117	A1	10/2004	Hertling et al.	
2004/0205260	A1	10/2004	Oki et al.	
2005/0012723	A1	1/2005	Pallakoff	
2005/0088410	A1	4/2005	Chaudhri	345/157
2005/0091359	A1	4/2005	Soin et al.	
2005/0101300	A1	5/2005	Hon et al.	
2005/0114797	A1	5/2005	Nguyen	
2005/0120312	A1	6/2005	Nguyen	
2005/0184145	A1	8/2005	Law et al.	
2005/0246662	A1	11/2005	Torrey et al.	715/841
2005/0267935	A1	12/2005	Gandhi et al.	
2005/0278708	A1	12/2005	Zhao et al.	
2006/0026535	A1	2/2006	Hotelling et al.	
2006/0033724	A1	2/2006	Chaudhri et al.	345/173
2006/0068769	A1	3/2006	Adya et al.	
2006/0179118	A1 *	8/2006	Stirbu	709/217
2006/0184614	A1	8/2006	Baratto et al.	
2006/0206827	A1	9/2006	DeWitt	
2006/0288306	A1	12/2006	Mahajan et al.	
2007/0008922	A1	1/2007	Abhishek et al.	
2007/0050470	A1	3/2007	Suzuki et al.	
2007/0056009	A1	3/2007	Spilo et al.	
2007/0061460	A1	3/2007	Khan et al.	
2007/0067328	A1	3/2007	Mingot et al.	
2007/0074120	A1	3/2007	Lindhorst	
2007/0140189	A1	6/2007	Muhamed et al.	
2007/0150822	A1	6/2007	Mansour et al.	
2007/0156677	A1	7/2007	Szabo	
2007/0192329	A1	8/2007	Croft et al.	
2007/0234048	A1	10/2007	Ziv	
2007/0236470	A1	10/2007	Abanami et al.	
2007/0262964	A1	11/2007	Zotov et al.	
2007/0288640	A1	12/2007	Schmieder	
2008/0016467	A1	1/2008	Chambers et al.	
2008/0088602	A1	4/2008	Hotelling	
2008/0155012	A1	6/2008	Chiang	
2008/0222416	A1	9/2008	Kiwimagi et al.	
2008/0222618	A1	9/2008	Valtchev	
2008/0291210	A1	11/2008	Partani et al.	
2008/0316218	A1	12/2008	Kilani et al.	
2009/0016529	A1	1/2009	Gopinath et al.	
2009/0055751	A1 *	2/2009	DePue et al.	715/740
2009/0058822	A1	3/2009	Chaudhri	
2009/0083628	A1	3/2009	Fitzmaurice et al.	
2009/0089667	A1	4/2009	Wood et al.	715/273
2009/0094523	A1	4/2009	Treder et al.	
2009/0100129	A1 *	4/2009	Vigil	G06F 3/0227 709/203
2009/0183200	A1	7/2009	Gritton et al.	725/37
2009/0193340	A1	7/2009	Mahajan et al.	
2009/0216683	A1	8/2009	Gutierrez	
2009/0238204	A1	9/2009	Kipnis	
2009/0319403	A1	12/2009	Schlueter	
2009/0322687	A1	12/2009	Duncan et al.	
2009/0327918	A1	12/2009	Aaron et al.	
2010/0005395	A1	1/2010	Shirakawa	
2010/0011299	A1	1/2010	Brodersen et al.	
2010/0079459	A1	4/2010	Breeds	
2010/0082733	A1 *	4/2010	Bernstein et al.	709/203
2010/0099396	A1 *	4/2010	Huq et al.	455/420
2010/0138780	A1	6/2010	Marano et al.	
2010/0174713	A1	7/2010	Baessler et al.	
2010/0177645	A1	7/2010	Kang et al.	
2010/0180190	A1	7/2010	Carroll	
2010/0211882	A1	8/2010	Bailey et al.	
2010/0250903	A1	9/2010	Jensen	
2010/0299436	A1	11/2010	Khalid et al.	
2010/0323762	A1	12/2010	Sindhu	
2011/0029896	A1	2/2011	Cheng	
2011/0093822	A1	4/2011	Sherwani	
2011/0099497	A1	4/2011	Fok et al.	
2011/0145696	A1 *	6/2011	Chiang	G06F 17/30905 715/234
2011/0161831	A1	6/2011	Karaoguz et al.	
2011/0219313	A1	9/2011	Mazzaferri	
2011/0239127	A1	9/2011	Meng et al.	
2011/0246904	A1	10/2011	Pinto et al.	

(56) **References Cited**

U.S. PATENT DOCUMENTS

2012/0317238	A1	12/2012	Beard
2013/0018765	A1	1/2013	Fork
2013/0182951	A1	7/2013	Shustorovich et al.

FOREIGN PATENT DOCUMENTS

WO	WO 99/63430	12/1999
WO	2005/112384	11/2005
WO	WO 2008/087409 A1	7/2008

OTHER PUBLICATIONS

"Jaadu," http://www.creativeapplications.net/iphone/jaadu-vnc-iphone/, Dec. 26, 2008, pp. 1-5.

"TiPb at Work: Jaadu VNC vs Mocha VNC," http://www.tipb.com/2008/10/22/tipb-at-work-jaadu-vnc-vs-mocha-vnc/, Oct. 22, 2008, pp. 1-6.

Extended European Search Report; Application No. 10764910.5-2211; pp. 7, Oct. 13, 2012.

Extended European Search Report; Application No. 10764911.3-2211; pp. 7, Oct. 13, 2012.

Session Initiation Protocol (SIP) Basic Call Flow Examples RFC 3665, pp. 1-94, Dec. 2003.

Extended European Search Report; Application No. 10764908.9-2211; pp. 7, Oct. 31, 2012.

Extended European Search Report; Application No. 10764913.9-2211; pp. 8, Nov. 26, 2012.

* cited by examiner

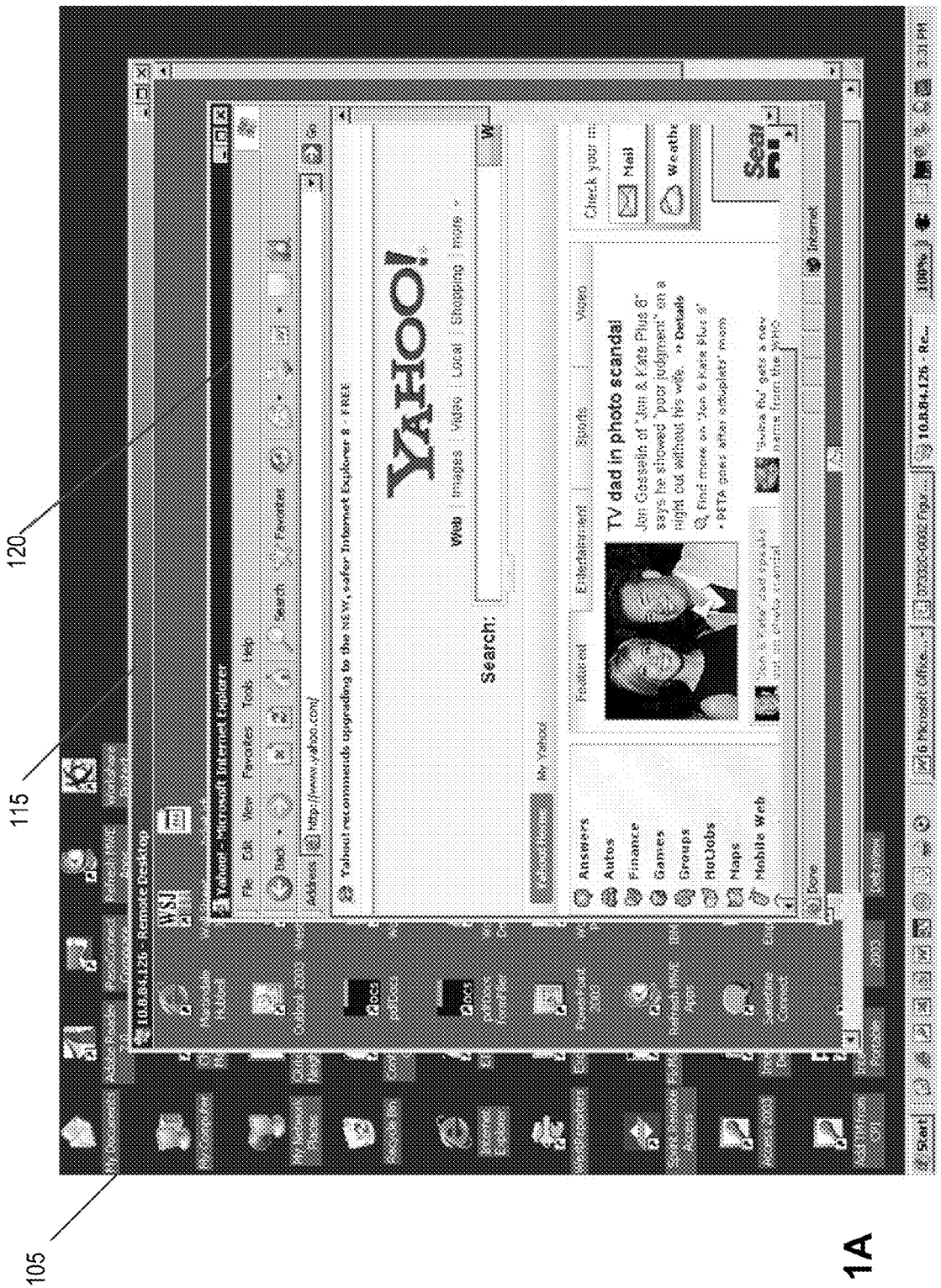

100
105
115
120
YAHOO!
Search:
TV dad in photo scandal
Answers
Autos
Finance
Games
Groups
HotJobs
Maps
Mobile Web

FIG. 1A

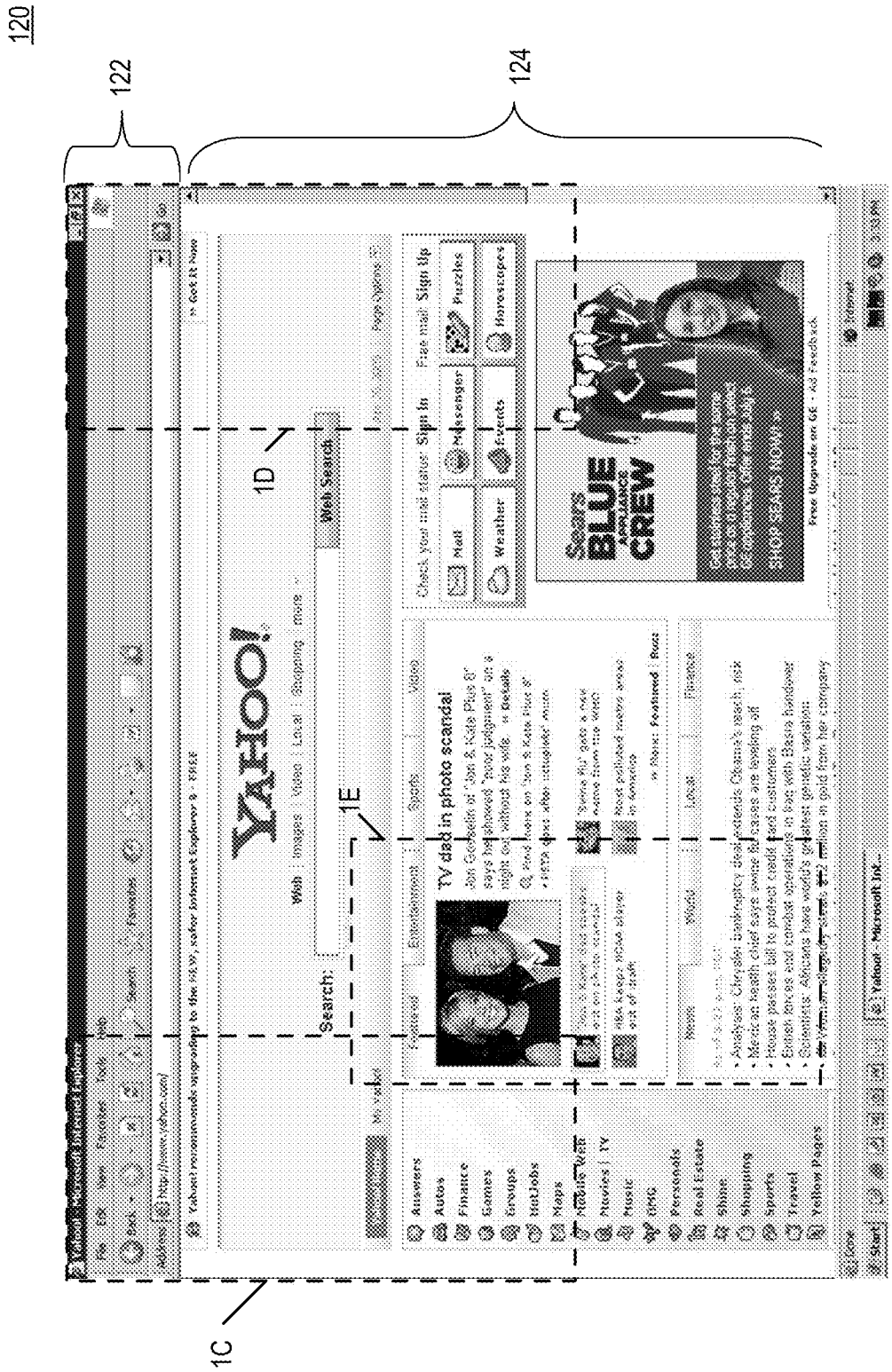

120
122
124
1C
1D
1E
YAHOO!
Search:
Web Search
TV dad in photo scandal
Sears
BLUE
APPLIANCE
CREW

FIG. 1B

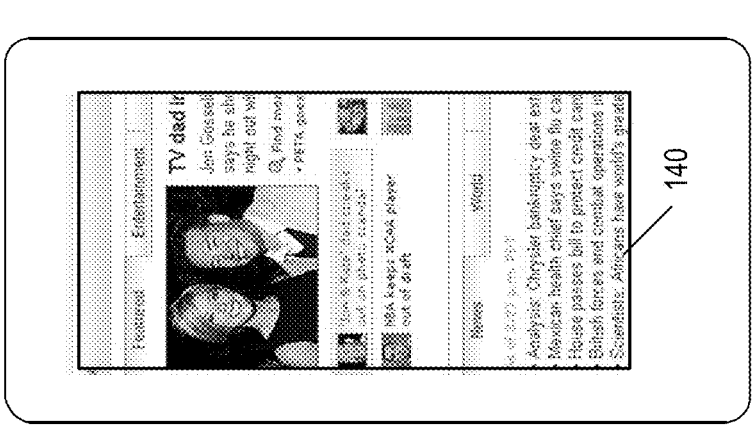


FIG. 1E

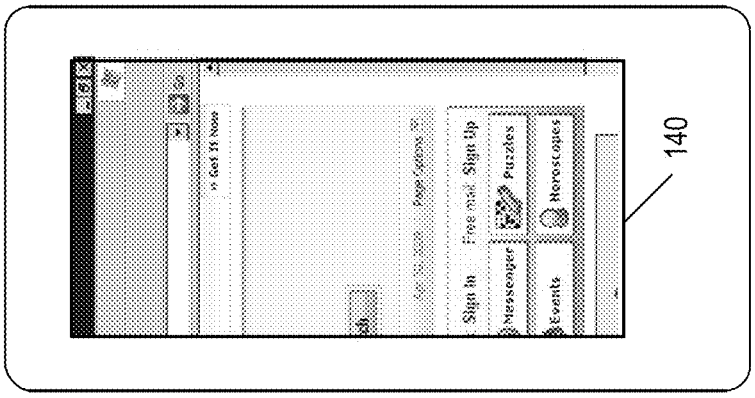


FIG. 1D

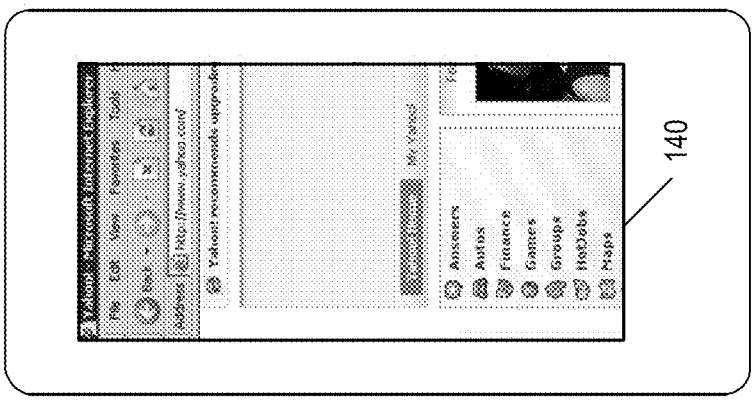


FIG. 1C

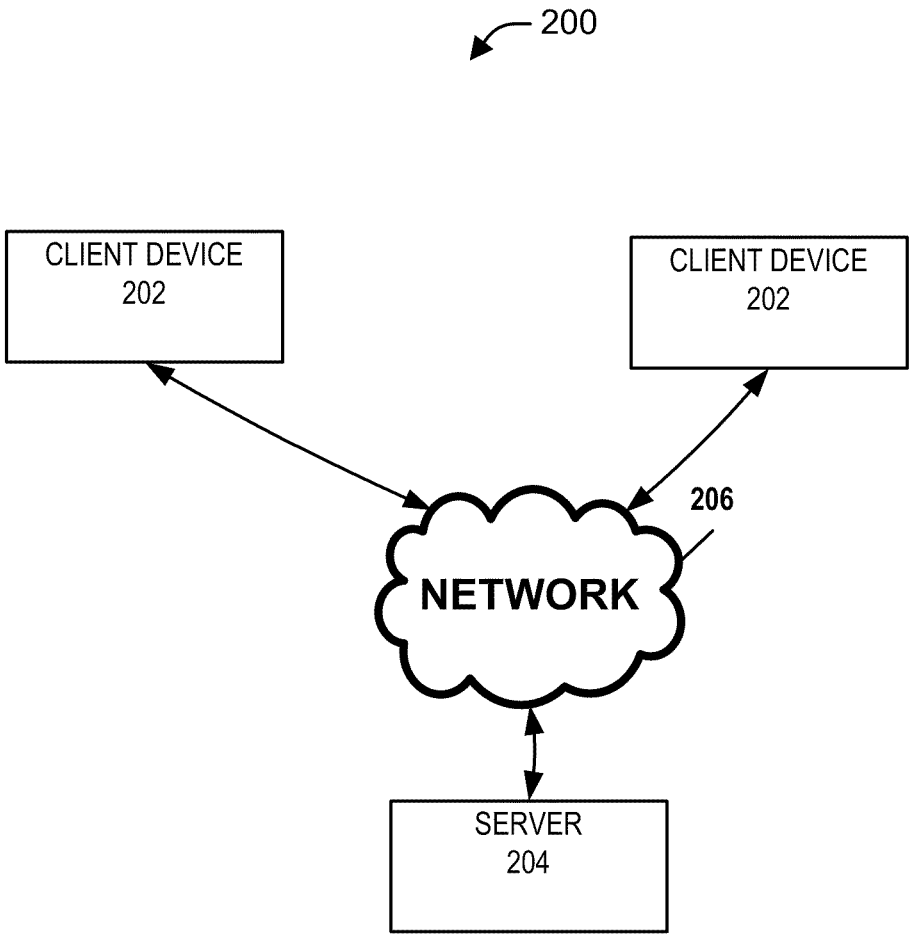

200
CLIENT DEVICE
202
CLIENT DEVICE
202
206
NETWORK
SERVER
204

FIG. 2

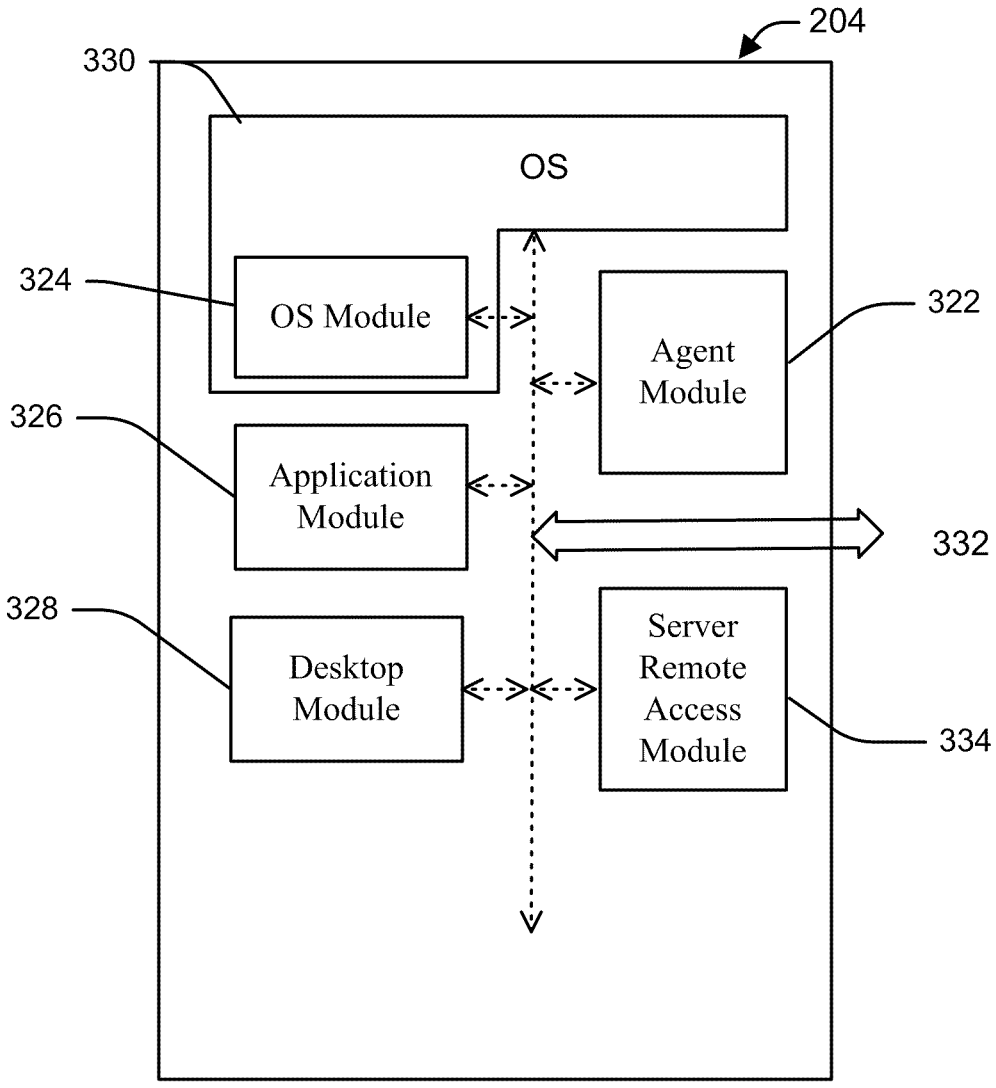

204
330
OS
324
OS Module
322
Agent Module
326
Application Module
332
328
Desktop Module
Server Remote Access Module
334

FIG. 3

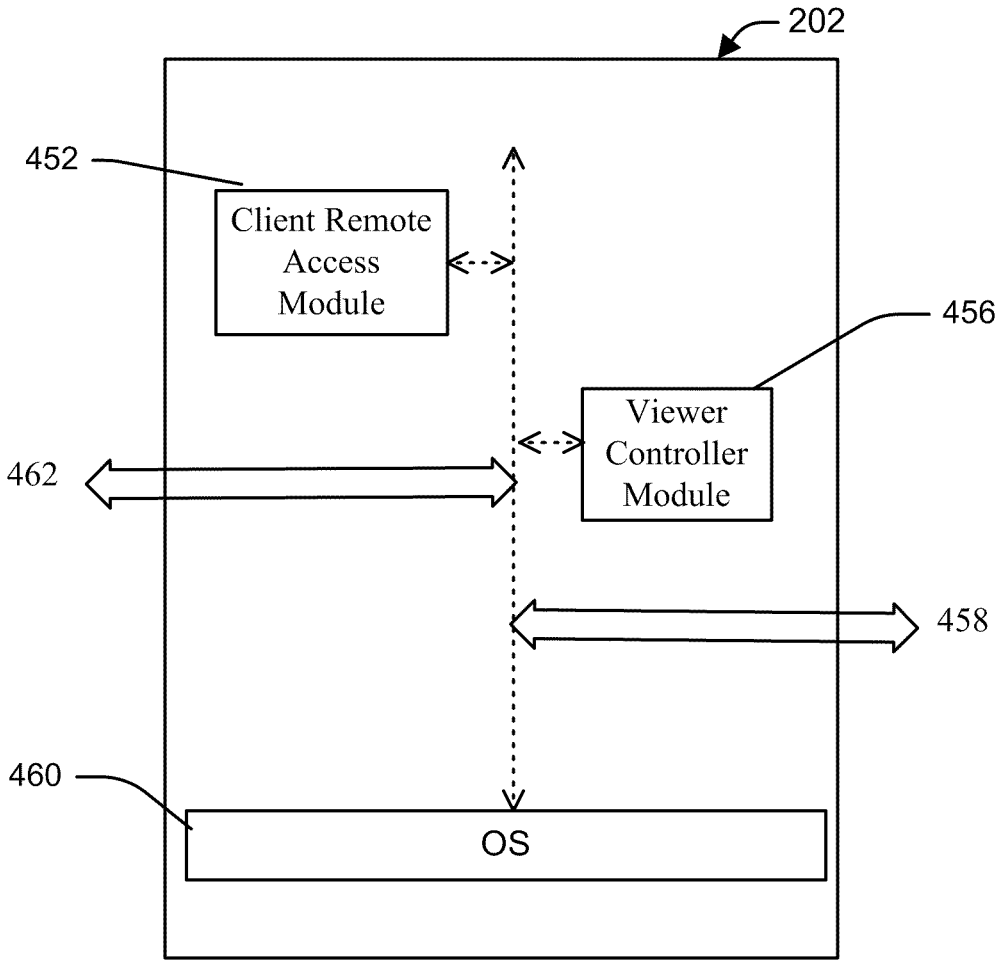
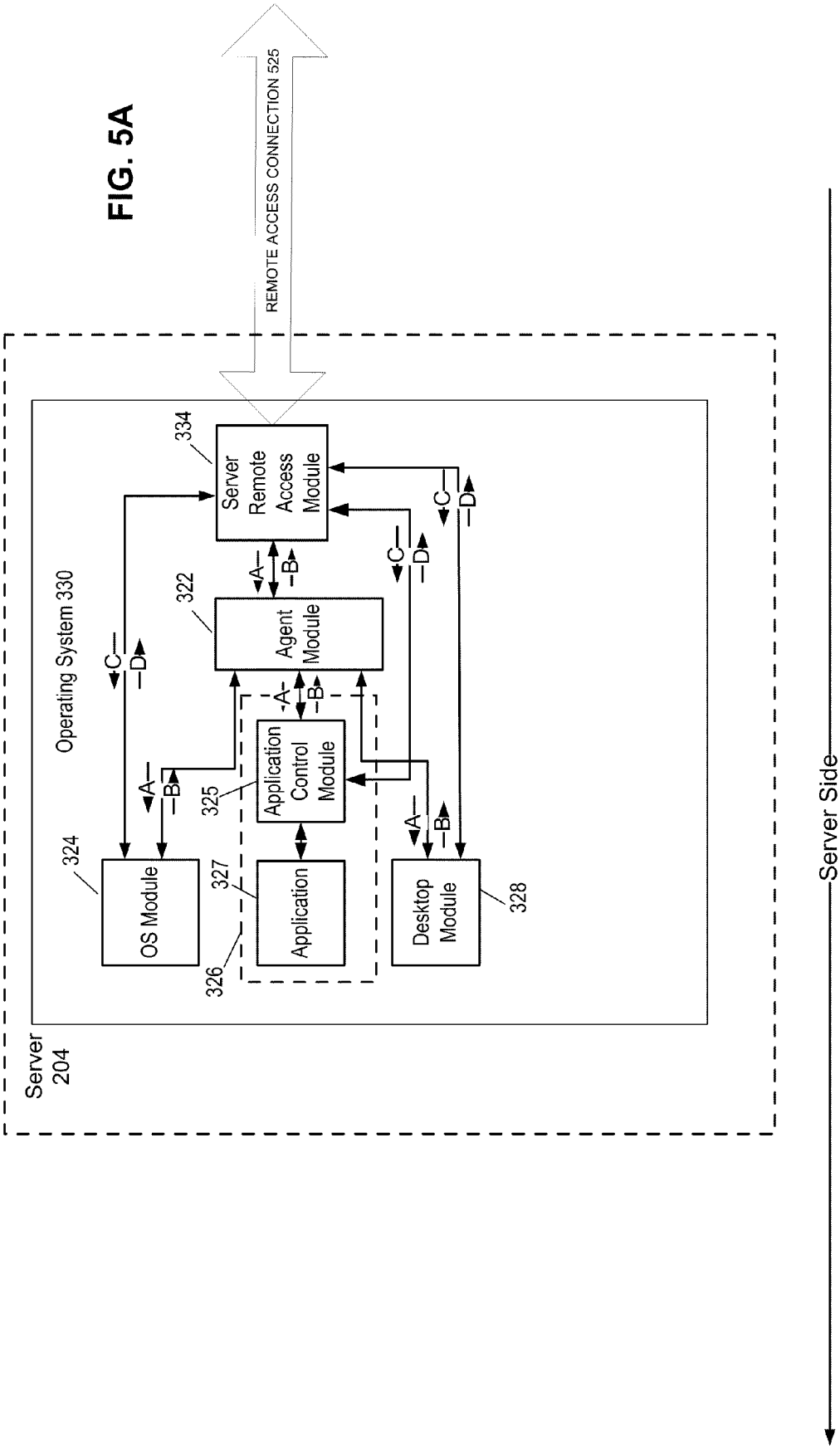

202
452
Client Remote Access Module
456
Viewer Controller Module
462
458
460
OS

FIG. 4

FIG. 5A


REMOTE ACCESS CONNECTION 525
Server Side
Server 204
Operating System 330
334
Server Remote Access Module
322
Agent Module
325
Application Control Module
324
OS Module
327
Application
326
328
Desktop Module
A
B
C
D

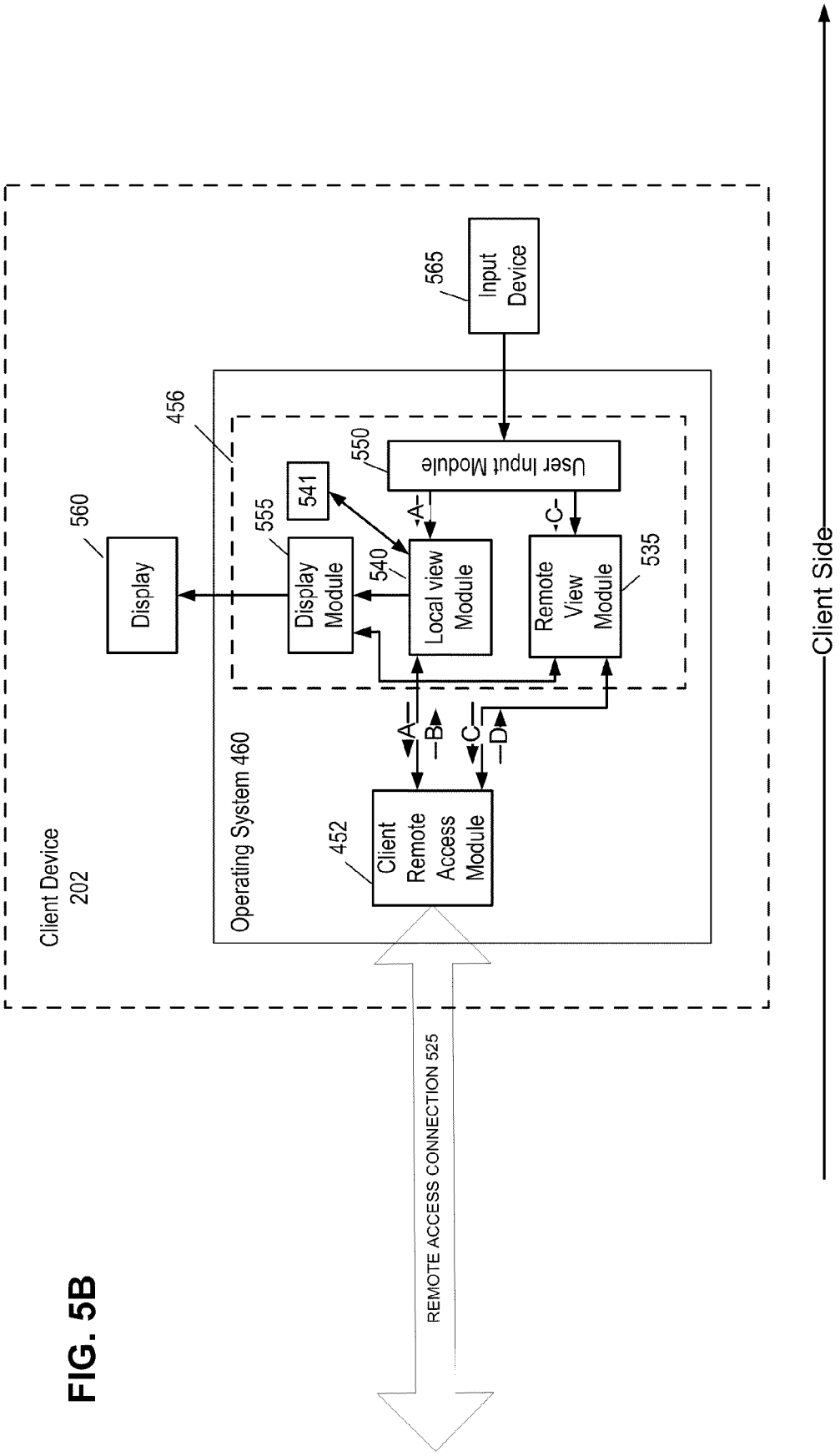

Client Side
Client Device
202
Operating System 460
456
Input Device
565
User Input Module
550
541
Display
560
Display Module
555
Local view Module
540
Remote View Module
535
Client Remote Access Module
452
A
B
C
D
REMOTE ACCESS CONNECTION 525

FIG. 5B

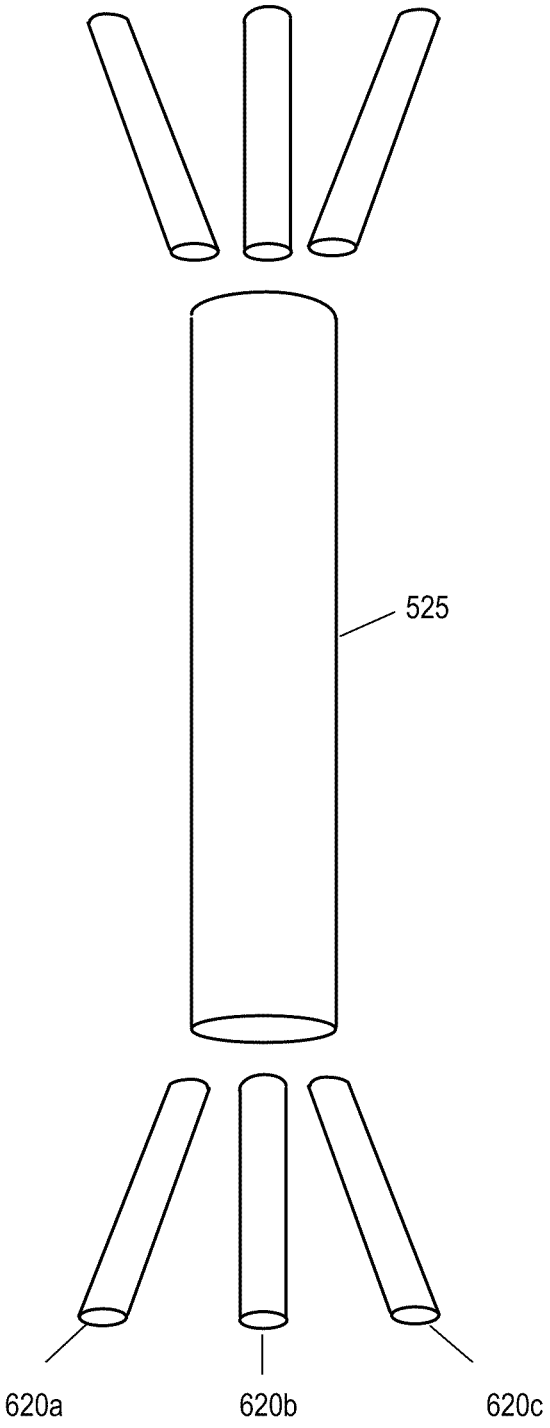


FIG. 6

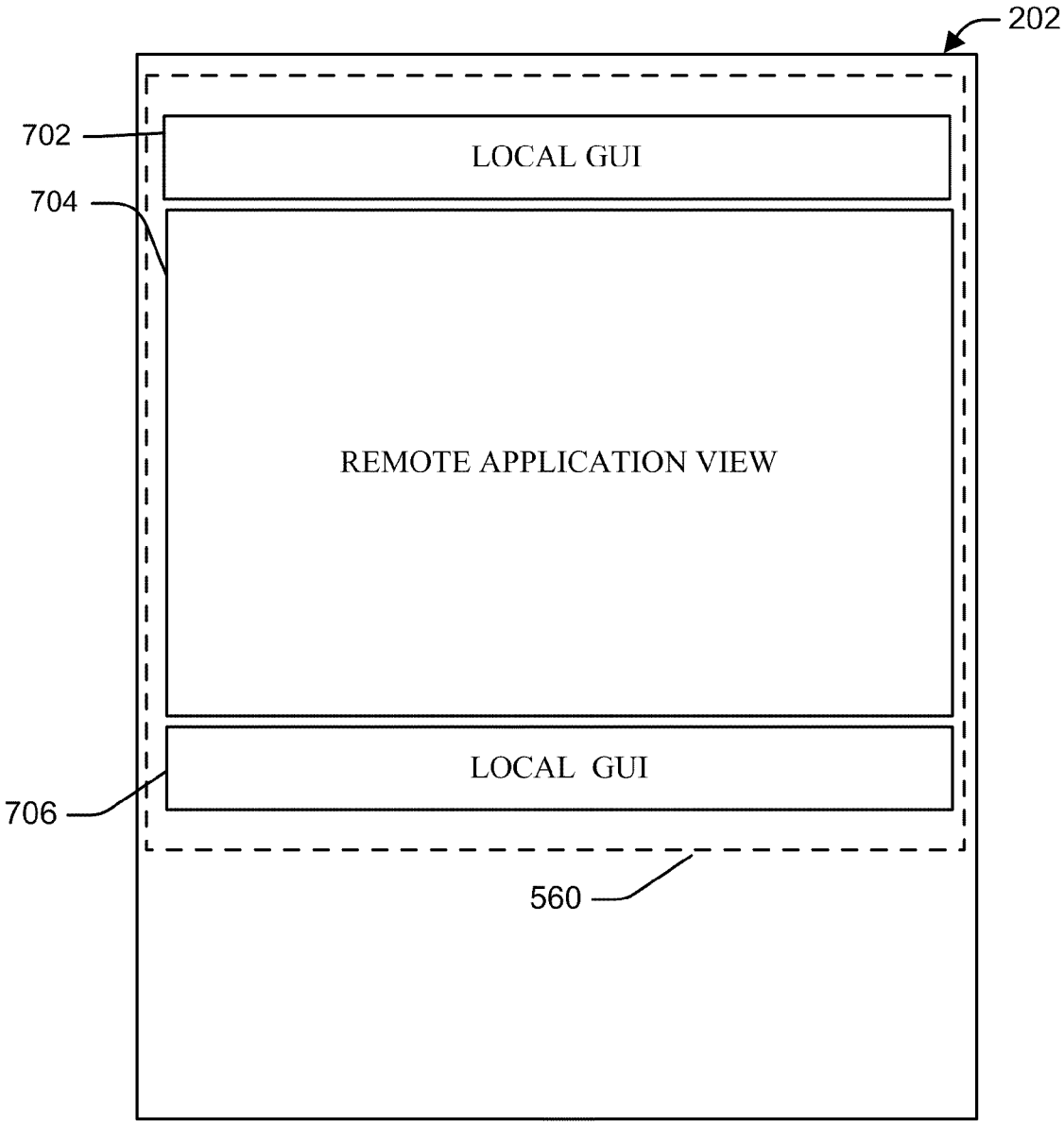

202
702
LOCAL GUI
704
REMOTE APPLICATION VIEW
LOCAL GUI
706
560

FIG. 7

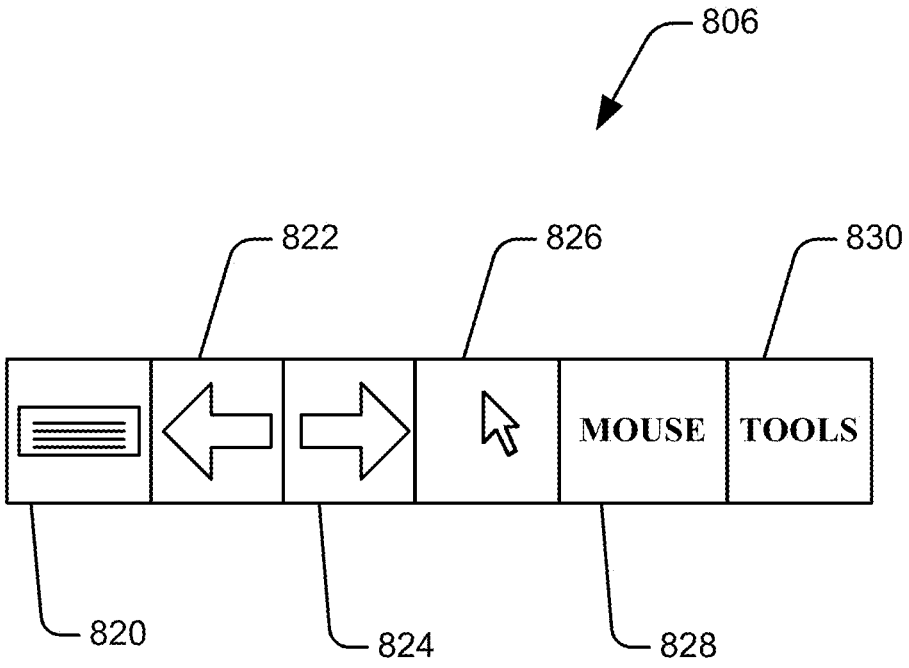

806
822
826
830
MOUSE
TOOLS
820
824
828

FIG. 8A

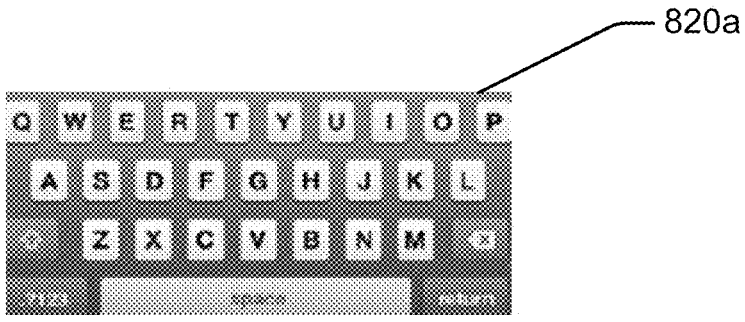

820a
Q W E R T Y U I O P
A S D F G H J K L
Z X C V B N M

FIG. 8B

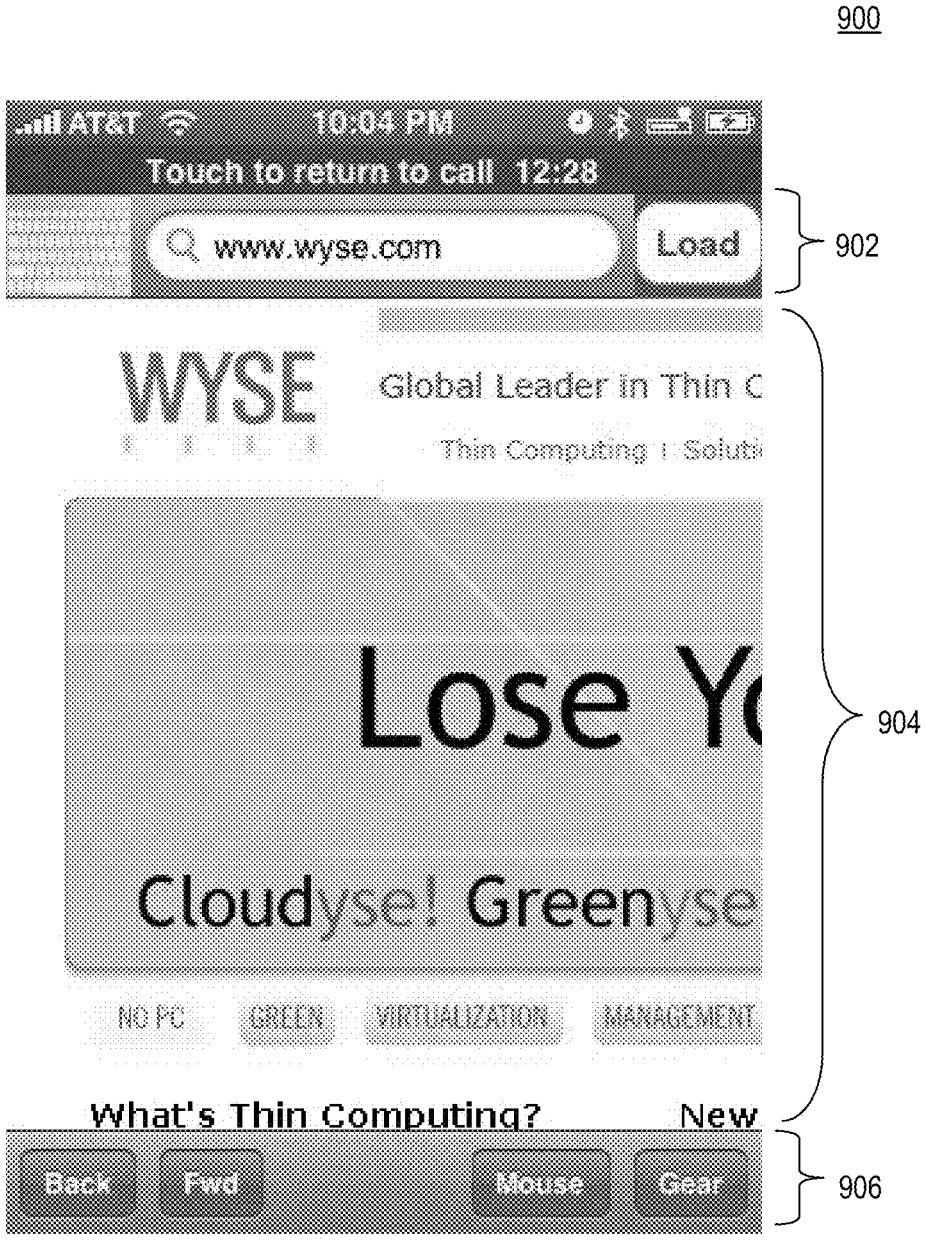

900
AT&T
10:04 PM
Touch to return to call 12:28
www.wyse.com
Load
902
WYSE
Global Leader in Thin C
Thin Computing | Soluti
Lose Yo
Cloudyse! Greenyse
NO PC
GREEN
VIRTUALIZATION
MANAGEMENT
What's Thin Computing?
New
904
Back
Fwd
Mouse
Gear
906

FIG. 9

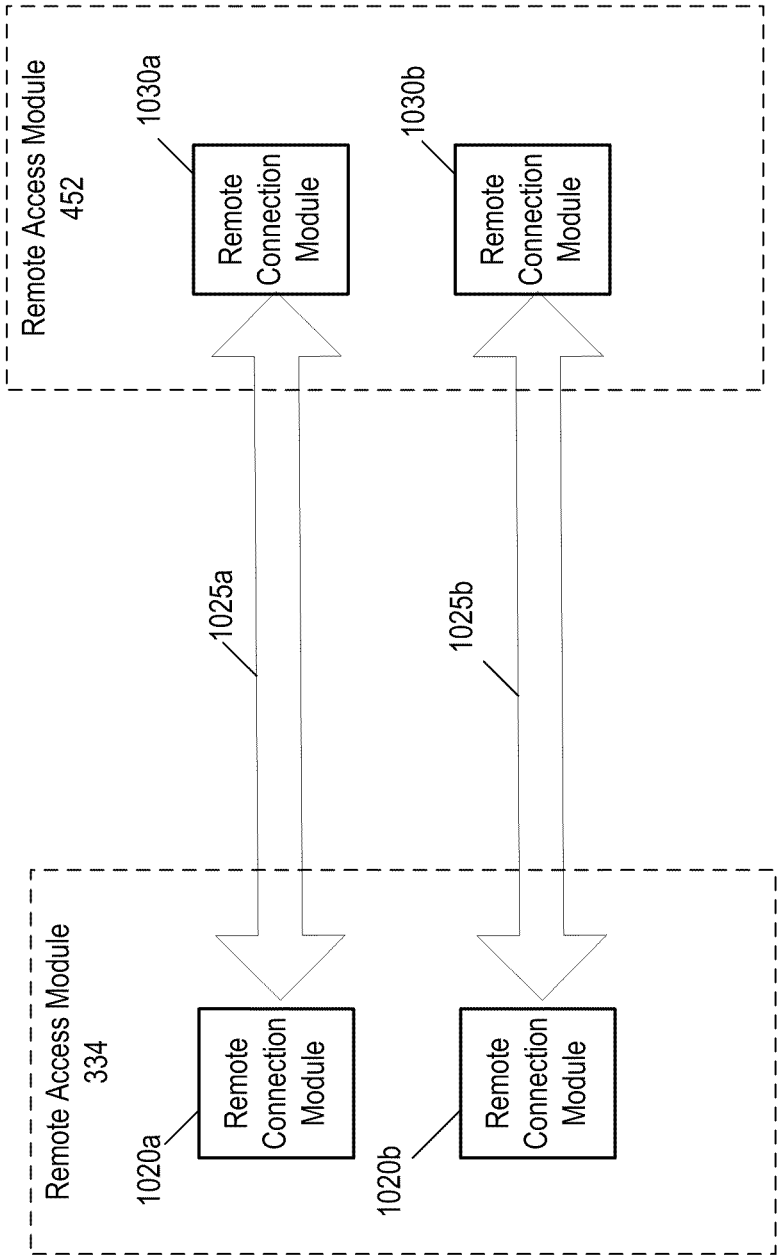

Remote Access Module
452
1030a
Remote Connection Module
1030b
Remote Connection Module
1025a
1025b
Remote Access Module
334
1020a
Remote Connection Module
1020b
Remote Connection Module

FIG. 10

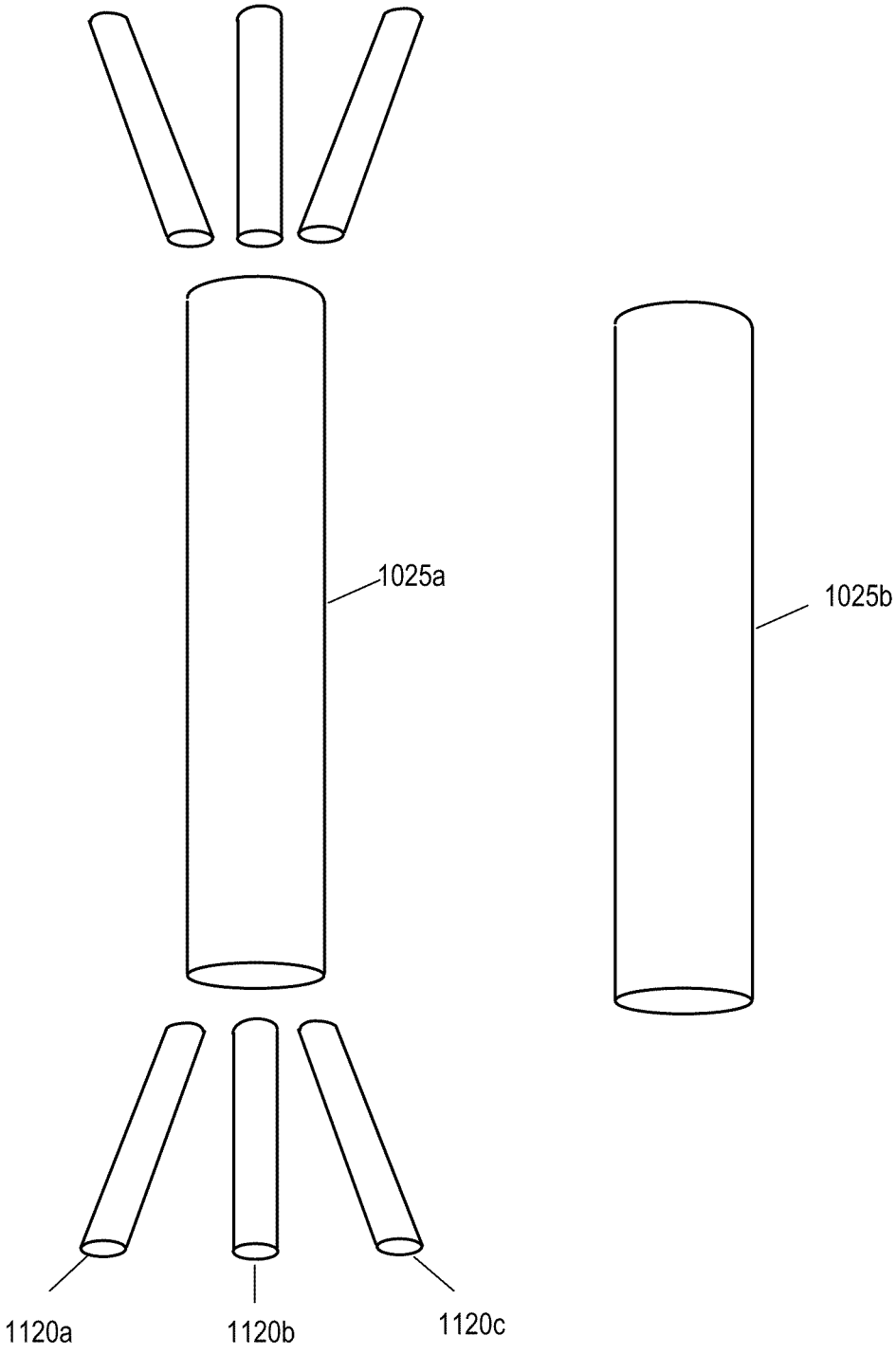


FIG. 11

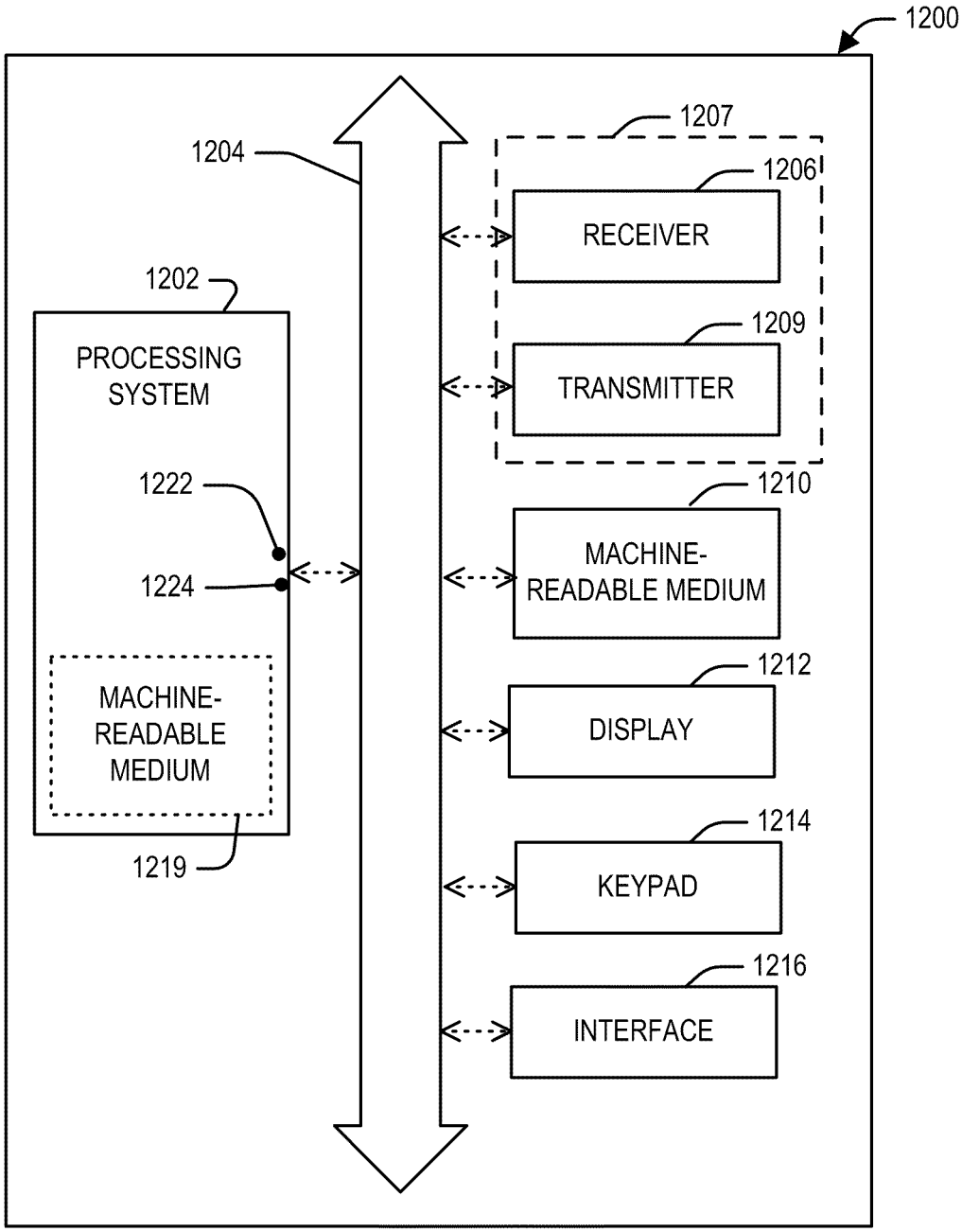

1200
1204
1207
1206
RECEIVER
1202
PROCESSING SYSTEM
1209
TRANSMITTER
1222
1210
MACHINE-READABLE MEDIUM
1224
MACHINE-READABLE MEDIUM
1212
DISPLAY
1214
KEYPAD
1219
1216
INTERFACE

FIG. 12

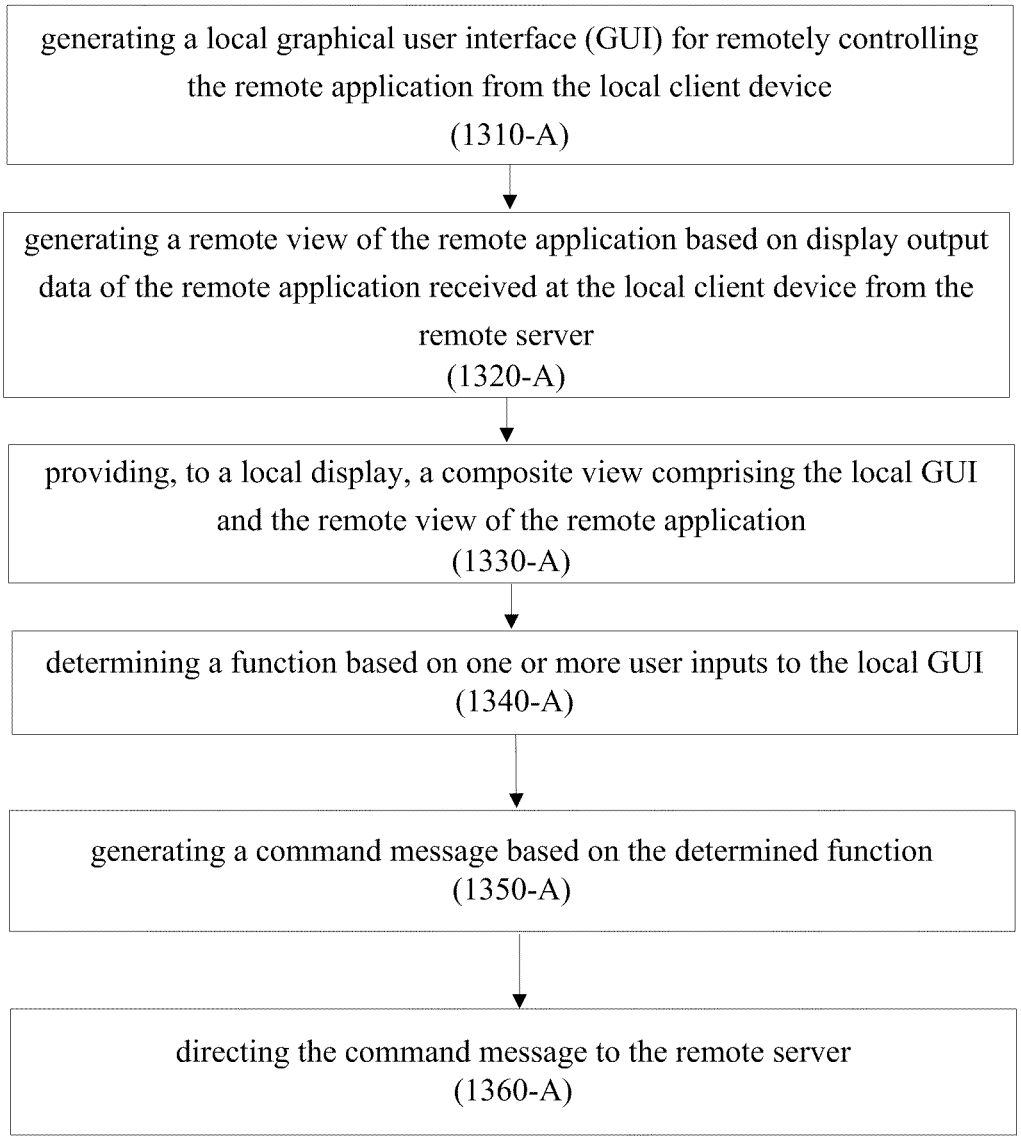

generating a local graphical user interface (GUI) for remotely controlling the remote application from the local client device
(1310-A)
generating a remote view of the remote application based on display output data of the remote application received at the local client device from the remote server
(1320-A)
providing, to a local display, a composite view comprising the local GUI and the remote view of the remote application
(1330-A)
determining a function based on one or more user inputs to the local GUI
(1340-A)
generating a command message based on the determined function
(1350-A)
directing the command message to the remote server
(1360-A)

FIG. 13A

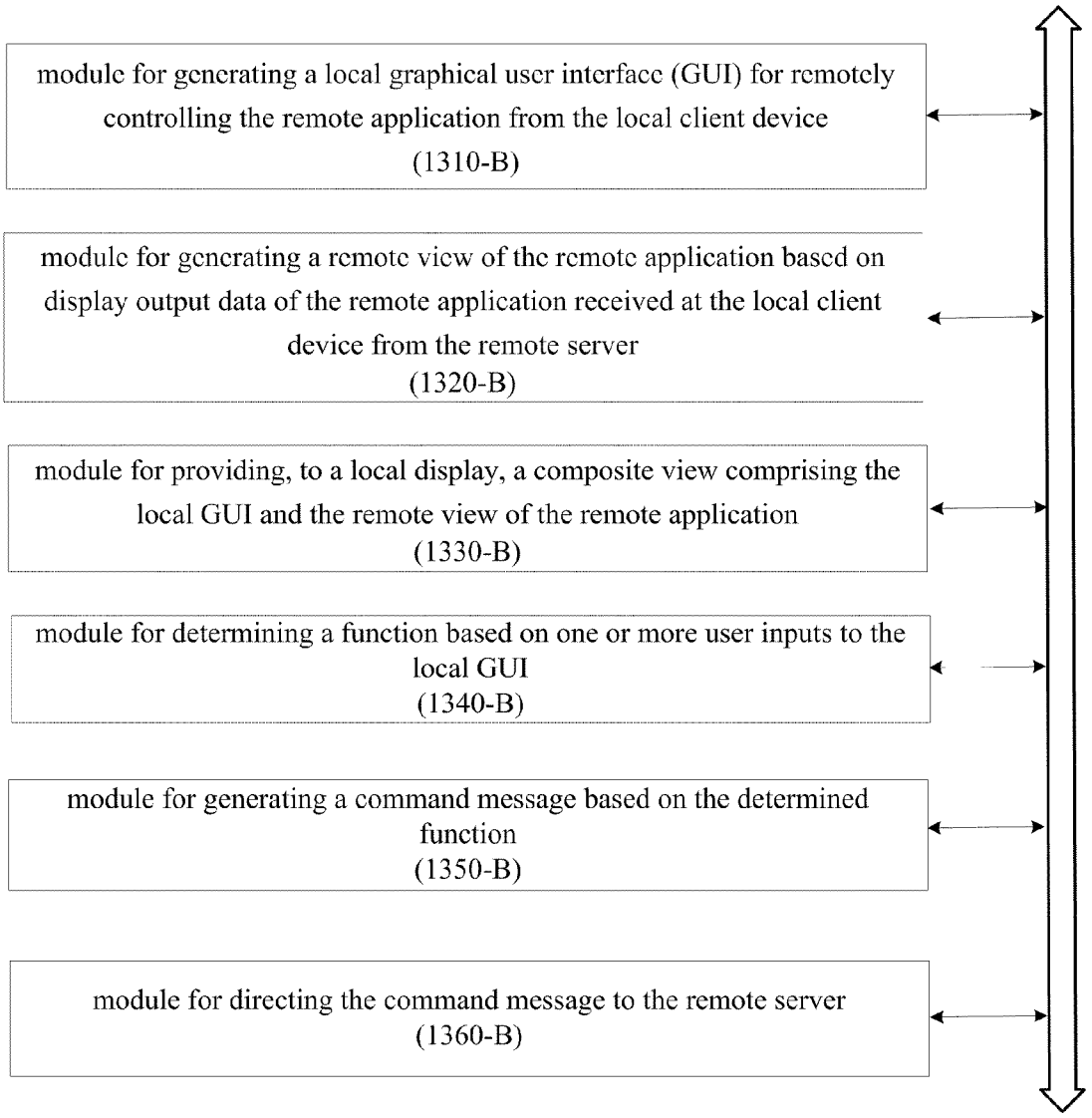

module for generating a local graphical user interface (GUI) for remotely controlling the remote application from the local client device
(1310-B)
module for generating a remote view of the remote application based on display output data of the remote application received at the local client device from the remote server
(1320-B)
module for providing, to a local display, a composite view comprising the local GUI and the remote view of the remote application
(1330-B)
module for determining a function based on one or more user inputs to the local GUI
(1340-B)
module for generating a command message based on the determined function
(1350-B)
module for directing the command message to the remote server
(1360-B)

FIG. 13B

SERVER-SIDE COMPUTING FROM A REMOTE CLIENT DEVICE

CROSS-REFERENCES TO RELATED APPLICATIONS

The present application claims the benefit of priority under 35 U.S.C. §119 from U.S. Provisional Patent Application Ser. No. 61/169,664, entitled "ENABLING SERVER SIDE COMPUTING ON A REMOTE CLIENT WHILE FACILITATING AN IMPROVED USER EXPERIENCE FOR THE REMOTE CLIENT USER," filed on Apr. 15, 2009, which is hereby incorporated by reference in its entirety for all purposes.

BACKGROUND

A client device may access remotely an application running on a remote server. However, a remote application may be designed for a desktop environment for a large screen, which can make it very difficult for a user at a client device having a small screen to interact with the remote application.

Instead of remotely accessing a desktop application from a client device, the application may be rewritten to run on the client device. However, client devices are typically slower and have limited resources compared with servers (e.g., desktop computer). As a result, rewriting an application on a server to run on a client device can be a huge undertaking requiring a large investment and long development time. For example, it can be very difficult to port existing code from an operating system of a server to a client device (e.g., smartphone). In many cases it is not practical to rewrite an application from scratch to match the native functionality and user interface of a client device.

Accordingly, there is a need for systems and methods that facilitate user interaction with an application running on a server from a client device and provide an improved experience for users at the client device.

SUMMARY

In one aspect of the disclosure, a system may be provided for rendering at a local client side a composite view including a local graphical user interface and a remote application view associated with a remote application running at a remote server. The system may comprise a local view module configured to do one or more of the following: generate a local graphical user interface (GUI) for controlling the remote application remotely from the local client side, receive one or more user inputs to the local GUI, determine a function corresponding to the one or more user inputs, generate a command message based on the determined function, and direct the command message to the remote server. The system may comprise a remote view module configured to generate a remote application view of the remote application based on display output data of the remote application received at the local client side from the remote server. The system may comprise a display module configured to provide, to a local display at the local client side, a composite view. The composite view may comprise the local GUI and the remote application view of the remote application.

In another aspect of the disclosure, a method may be provided for rendering at a local client side a composite view including a local graphical user interface and a remote application view associated with a remote application running at a remote server. The method may comprise one or more of the following: generating a local graphical user interface (GUI) for remotely controlling the remote application from the local client device, generating a remote view of the remote application based on display output data of the remote application received at the local client device from the remote server, providing, to a local display, a composite view comprising the local GUI and the remote view of the remote application, determining a function based on one or more user inputs to the local GUI, generating a command message based on the determined function, and directing the command message to the remote server.

In yet another aspect of the disclosure, a machine-readable medium encoded with instructions may be provided for rendering at a local client side a composite view including a local graphical user interface and a remote application view associated with a remote application running at a remote server. The instructions may comprise code for one or more of the following: generating a local graphical user interface (GUI) for remotely controlling the remote application from the local client device, generating a remote view of the remote application based on display output data of the remote application received at the local client device from the remote server, providing, to a local display, a composite view comprising the local GUI and the remote view of the remote application, determining a function based on one or more user inputs to the local GUI, generating a command message based on the determined function, and directing the command message to the remote server.

In yet another aspect of the disclosure, an apparatus may be provided for rendering at a local client side a composite view including a local graphical user interface and a remote application view associated with a remote application running at a remote server. The apparatus may comprise one or more of the following: means for generating a local graphical user interface (GUI) for remotely controlling the remote application from the local client device, means for generating a remote view of the remote application based on display output data of the remote application received at the local client device from the remote server, means for providing, to a local display, a composite view comprising the local GUI and the remote view of the remote application, means for determining a function based on one or more user inputs to the local GUI, means for generating a command message based on the determined function, and means for directing the command message to the remote server.

It is understood that other configurations of the subject technology will become readily apparent to those skilled in the art from the following detailed description, wherein various configurations of the subject technology are shown and described by way of illustration. As will be realized, the subject technology is capable of other and different configurations and its several details are capable of modification in various other respects, all without departing from the scope of the subject technology. Accordingly, the drawings and detailed description are to be regarded as illustrative in nature and not as restrictive.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1**A illustrates an example of a display on a client device during a remote access session with a remote server.

FIG. **1**B illustrates an example of a web browser running on a server.

FIGS. **1**C to **1**E illustrate examples of different portions of the web browser displayed on a client device.

FIG. **2** is a conceptual block diagram of a computer network according to certain aspects of the present disclosure.

FIG. **3** is a conceptual block diagram of a server according to certain aspects of the present disclosure.

FIG. **4** is a conceptual block diagram of a client device according to certain aspects of the present disclosure.

FIG. **5**A is a conceptual block diagram of a server according to certain aspects of the present disclosure.

FIG. **5**B is a conceptual block diagram of a client according to certain aspects of the present disclosure.

FIG. **6** illustrates an example of a remote access connection.

FIG. **7** illustrates a composite view at the client device according to certain aspects of the present disclosure.

FIG. **8**A illustrates an example of a local GUI according to certain aspects of the present disclosure.

FIG. **8**B illustrates an example of a graphical keyboard according to certain aspects of the present disclosure.

FIG. **9** illustrates a display at the client device according to certain aspects of the present disclosure.

FIG. **10** is a conceptual block diagram of server and client remote access modules according to certain aspects of the present disclosure.

FIG. **11** illustrates an example of a plurality of connections between the server and the client device.

FIG. **12** is a block diagram that illustrates an exemplary computing system in accordance with certain embodiments of the present disclosure.

FIG. **13**A illustrates an example of an operation of a system according to one aspect of the present disclosure.

FIG. **13**B illustrates an example of a configuration of an apparatus according to one aspect of the present disclosure.

DETAILED DESCRIPTION

The detailed description set forth below is intended as a description of various configurations of the subject technology and is not intended to represent the only configurations in which the subject technology may be practiced. The appended drawings are incorporated herein and constitute a part of the detailed description. The detailed description includes specific details for the purpose of providing a thorough understanding of the subject technology. However, it will be apparent to those skilled in the art that the subject technology may be practiced without these specific details. In some instances, well-known structures and components are shown in block diagram form in order to avoid obscuring the concepts of the subject technology. Like components are labeled with identical element numbers for ease of understanding.

FIG. **1**A shows an example of a display **100** at a client device with a large screen during a remote access session with a server. The display **100** includes the local desktop **105** of the client device, a remote view window **115** showing an image of a remote desktop **115** and a remote web browser application **120** running on the server. The image of the remote desktop **115** and remote application **120** are based on display output data of the remote desktop **115** and remote application **120** from the server.

FIG. **1**B shows an enlarged view of the remote web browser application **120**. The remote web browser application **120** includes a GUI **122** (e.g., toolbars) and a window **124** for displaying web content. The GUI **122** is designed to be viewed on a large display. As a result, it can be very different for a user at a client device with a small display to remotely interact with the remote web browser **120**. For example, if the image of the entire remote web browser application **120** is displayed on a client device with a small display, then the GUI **122** is greatly reduced in size making it difficult for the user to view the GUI **122** and select graphical control objects in the GUI **122**.

If the user at the client device zooms in on a portion of the remote web browser application **120** to enlarge that portion for viewing on a client device with a small screen (e.g., mobile phone), then the user needs to constantly manipulate the remote view by adjusting the zoom and scrolling to view different portions of the web browser application **120**. FIG. **1**C shows an example of a client device **135** in which a portion of the remote web browser application **120** has been zoomed in for viewing on the display **140** of the client device **135** (e.g., smartphone). In this example, only a portion of the GUI **122** is visible. As a result, the user at the client device has to adjust the zoom and/or scroll the remote view on the display **140** to view the GUI **122**. FIG. **1**D shows another example in which a different portion of the GUI **122** is visible on the display **140**. FIG. **1**E shows an example in which the GUI **122** is not visible at all on the display **140**. The different portions of the web browser application **120** displayed in FIGS. **1**C to **1**E are shown in FIG. **1**B in dashed boxes. Thus, interacting with a remote application running on a server from a client device (e.g., smartphone) can be very cumbersome and difficult.

FIG. **2** illustrates a simplified diagram of a system **200** in accordance with an aspect of the present disclosure. The system **200** may include one ore more remote client devices **202** in communication with a server computing device **204** (server) via a network **206**. In one aspect, the server **204** is configured to allow remote sessions (e.g., remote desktop sessions) wherein users can access applications and files on the server **204** by logging onto the server **204** from a client device **202**. Such a connection may be established using any of several well-known techniques such as the Remote Desktop Protocol (RDP) on a Windows-based server.

By way of illustration and not limitation, in one aspect of the disclosure, stated from a perspective of a server side (treating a server as a local device and treating a client device as a remote device), a server application is executed (or runs) at a server **204**. While a remote client device **202** may receive and display a view of the server application on a display local to the remote client device **202**, the remote client device **202** does not execute (or run) the server application at the remote client device **202**. Stated in another way from a perspective of the client side (treating a server as remote device and treating a client device as a local device), a remote application is executed (or runs) at a remote server **204**.

By way of illustration and not limitation, a client device **202** can represent a computer, a mobile phone, a laptop computer, a thin client device, a personal digital assistant (PDA), a portable computing device, or a suitable device with a processor. In one example, a client device **202** is a smartphone (e.g., iPhone, Android phone, Blackberry, etc.). In certain configurations, a client device **202** can represent an audio player, a game console, a camera, a camcorder, an audio device, a video device, a multimedia device, or a device capable of supporting a connection to a remote server. In one example, a client device **202** can be mobile. In another example, a client device **202** can be stationary. According to one aspect of the disclosure, a client device **202** may be a device having at least a processor and memory, where the total amount of memory of the client device **202** could be less than the total amount of memory in a server **204**. In one example, a client device **202** does not have a

hard disk. In one aspect, a client device **202** has a display smaller than a display supported by a server **204**. In one aspect, a client device may include one or more client devices.

In one aspect, a server **204** may represent a computer, a laptop computer, a computing device, a virtual machine (e.g., VMware® Virtual Machine), a desktop session (e.g., Microsoft Terminal Server), a published application (e.g., Microsoft Terminal Server) or a suitable device with a processor. In one aspect, a server **204** can be stationary. In another aspect, a server **204** can be mobile. In certain configurations, a server **204** may be any device that can represent a client device. In one aspect, a server **204** may include one or more servers.

In one example, a first device is remote to a second device when the first device is not directly connected to the second device. In one example, a first remote device may be connected to a second device over a communication network such as a Local Area Network (LAN), a Wide Area Network (WAN), and/or other network.

When a client device **202** and a server **204** are remote with respect to each other, a client device **202** may connect to a server **204** over a network **206**, for example, via a modem connection, a LAN connection including the Ethernet or a broadband WAN connection including DSL, Cable, T1, T3, Fiber Optics, Wi-Fi, or a mobile network connection including GSM, GPRS, 3G, WiMax or other network connection. A network **206** can be a LAN network, a WAN network, a wireless network, the Internet, an intranet or other network. A network **206** may include one or more routers for routing data between client devices and/or servers. A remote device (e.g., client device, server) on a network may be addressed by a corresponding network address, such as, but not limited to, an Internet protocol (IP) address, an Internet name, a Windows Internet name service (WINS) name, a domain name or other system name. These illustrate some examples as to how one device may be remote to another device. But the subject technology is not limited to these examples.

According to certain aspects of the present disclosure, the terms "server" and "remote server" are generally used synonymously in relation to a client device, and the word "remote" may indicate that a server is in communication with other device(s), for example, over a network connection(s).

According to certain aspects of the present disclosure, the terms "client device" and "remote client device" are generally used synonymously in relation to a server, and the word "remote" may indicate that a client device is in communication with a server(s), for example, over a network connection(s).

In one aspect of the disclosure, a "client device" may be sometimes referred to as a client or vice versa. Similarly, a "server" may be sometimes referred to as a server device or vice versa.

In one aspect, the terms "local" and "remote" are relative terms, and a client device may be referred to as a local client device or a remote client device, depending on whether a client device is described from a client side or from a server side, respectively. Similarly, a server may be referred to as a local server or a remote server, depending on whether a server is described from a server side or from a client side, respectively. Furthermore, an application running on a server may be referred to as a local application, if described from a server side, and may be referred to as a remote application, if described from a client side.

In one aspect, devices placed on a client side (e.g., devices connected directly to a client device(s) or to one another using wires or wirelessly) may be referred to as local devices with respect to a client device and remote devices with respect to a server. Similarly, devices placed on a server side (e.g., devices connected directly to a server(s) or to one another using wires or wirelessly) may be referred to as local devices with respect to a server and remote devices with respect to a client device.

FIG. **3** illustrates a simplified block diagram of a server **204** in accordance with an aspect of the present disclosure. The server **204** comprises an agent module **322**, an OS module **324**, one or more application modules **326**, a desktop module **328**, a server remote access module **334** and an operating system (OS) **330** in communication with the modules. In one aspect, the OS module **324** can be a part of the OS **330**. The server **204** is communicatively coupled with the network **206** via a network interface **332**. The modules can be implemented in software, hardware and/or a combination of both. Features and functions of these modules according to various aspects are further described in the present disclosure.

FIG. **4** illustrates a simplified block diagram of a client device **202** in accordance with an aspect of the present invention. The client device **202** comprises a viewer controller (VC) module **456**, a client remote access module **452** and an operating system (OS) **460** in communication with the modules. The modules are further in communication with various user interface devices (not shown in FIG. **4**) via a human interface devices (HID) connection **458**. The user interface devices may include one or more output devices (e.g., one or more of a display, a speaker, or other audio, image or video output devices) and one or more input devices (e.g., one or more of a keyboard, a mouse, a trackball, a microphone, a stylus, a touch screen, a touch pad, a pen, a tablet, or other audio, image or video input devices). The modules are also in communication with the network **206** via a network connection **462**. The modules can be implemented in software, hardware and/or a combination of both. Additional features and functions of these modules according to various aspects of the present disclosure are further described in the disclosure.

FIGS. **5**A and **5**B are conceptual block diagrams illustrating a server **204** and a client device **202** according to an aspect of the disclosure.

Referring to FIG. **5**A, the server **204** may comprise the server remote access module **334** and the agent module **322**. The server remote access module **334** is configured to establish a remote access connection **525** with the client device **202** to provide the client device **202** with remote access to a desktop and/or applications running on the server **204**. The server remote access module **334** may comprise a remote access application and may communicate with the client device **202** over a network **206** based on a remote access protocol (e.g., RDP/ICA), other protocols or a combination of protocols. Remote access applications allow a user at a client device to remotely access a desktop and/or application running on a server. Examples of remote access applications include, but are not limited to, the Microsoft® Remote Desktop Protocol (RDP) application and the Citrix® Independent Computing Architecture (ICA) application.

The server remote access module **334** is also configured to receive command messages from the client device **202** and communicate the received command messages to the agent module **322**. The agent module **322** may be configured to control a desktop and/or application running on the server **204** based on the received command messages, as described further below.

The server **204** further comprises an application module **326**, the desktop module **328** and the OS module **324**. The application module **326** comprises an application **327** and an application control module **325**. An application **327** may include one or more applications. The application **327** may include, for example, a web browser application, a word processing application, a spreadsheet application, a game application, an audio application, and/or other applications. The application **327** may also include applications for rendering multi-media content including, for example, Adobe® Acrobat, Microsoft® Silverlight, and/or other applications. In one example, the application **327** is executed (or runs) at the server **204**, and the client device **202** does not execute (or run) the application **327** locally at the client device **202**.

An application control module **325** may include one or more application control modules. The application control module **325** may be configured to control the application **327** based on commands from the agent module **322**. The application control module **325** may comprise one or more ActiveX controllers for controlling Windows-based applications or other types of controller. The application control module **325** may also control the applications based on user inputs received by the remote access module **334** from the client device **202**. The user inputs may comprise pointer inputs, keyboard inputs and/or other inputs. The application control module **325** may also control the applications based on events generated by the application **327** or the OS **330**.

The desktop module **328** may be configured to generate a desktop that provides a GUI for allowing a user to select and launch applications on the server **204**, manage files, and configure certain features of the operating system **330**. The GUI may be a server-based GUI presented by the agent module **322**. Alternatively, the GUI may be controlled by the agent module **322** and displayed natively on the client device. The desktop may display icons representing different applications that can be launched on the server **204**, in which a user can select an application by moving a pointer to the corresponding icon and double clicking on the icon. The OS module **324** may be configured to communicate certain OS events to the agent module **322** and the server remote access module **334**, as described further below.

The server **204** further comprises the operating system **330**, which may manage and provide communication to and receive communication from the modules and application shown in FIG. **5**A. The operating system **330** may also manage multiple remote user sessions allowing multiple client devices to remotely access the server **204**. The operating system **330** may include Microsoft® Windows, Linux, Unix, Mac OS or another operating system. Although the server remote access module **334** is shown separately from the operating system **330**, the server remote access module **334** may be considered part of the operating system **330**. For example, the remote access module **334** may comprise an RDP application that comes pre-installed on the server **204** as part of Microsoft® Windows.

Referring now to FIG. **5**B, the client device **202** may comprise the client remote access module **452** and the viewer controller module **456**. The client remote access module **452** may be configured to communicate with the server **204** over a network **206** to remotely access a desktop and/or applications running on the server **204**. The client remote access module **452** may communicate with the server remote access module **334** based on a remote access protocol (e.g., RDP/ICA), other protocols or a combination of protocols. In one aspect, the client remote access module **452** may be configured to receive display output data of a desktop and/or application running on the server **204** from the server remote access module **334** over a network **206**.

In one aspect of the disclosure, display output data may comprise visual/audio information that a user located at a server **204** would have seen on a display at the server (e.g., a "virtual" display) and/or would have heard from an audio output such as a speaker at the server (e.g., a "virtual" speaker). In one aspect, the display output data may comprise an image (e.g., bitmap) and/or drawing commands of the display output of a desktop and/or application running on the server **204**. In one aspect, drawing commands provide instructions or information to a display system as to what is to be drawn on a display. The instructions or information may include, for example and without limitation, the location, texture, geometry, shading, light intensity, or color, or any combination therein, of a pixel or group of pixels of the display. In some aspects, a drawing command corresponds to multiple pixels. Still in some aspects, a drawing command corresponds to a single pixel of a display. In some aspects, drawing commands may include raster operations. In one aspect of the disclosure, the phrase "display output data" may be sometimes referred to as "display data", "display output" or vice versa.

The client remote access module **452** may be configured to send user inputs to the server remote access module **334** over a network **206**. The user inputs may comprise pointer inputs, keyboard inputs and/or other types of input. The client remote access module **452** may also be configured to send command messages to the server remote access module **334**, which are described in further detail below. The remote access connection **525** between the server **204** and client device **202** may comprise a plurality of virtual channels for communicating different types of data. For example, a RDP connection may include different virtual channels for display output data and user inputs. FIG. **6** illustrates an example of the remote access connection **525** comprising multiple virtual channels **620***a* to **620***c*.

The viewer controller module **456** may comprise a remote view module **535**, a local view module **540** and a display module **555**. The remote view module **535** may be configured to receive display output data of a desktop and/or application **327** running on the server **204** from the client remote access module **452** and generate a remote view of the display output data of the remote desktop and/or application. In one aspect of the disclosure, the phrase "remote view" may sometimes be referred to as "remote application view" or vice versa. In one aspect of the disclosure, the term "an application" or "a remote application" may refer to an application, a remote application, a desktop, or a remote desktop.

The local view module **540** may be configured to generate a local GUI that allows a user at the client device **202** to remotely control the application **327** running on the server **204**. The local view module **540** may generate the local GUI based on a configuration file **541**. In one aspect, the file **541** may be generated by the agent module **322** or application module **326** and be sent to the client device **202**. The file **541** may fully define the layout, design and logical operation of the local GUI. This would allow the server **204** to update the local GUI in the client device **202** without having to make modifications to the client device **202**. The local GUI may provide similar controls as the GUI of the application **327** running on the server **204**. The local GUI may include graphical control objects for controlling certain functions of the application **327**.

For the example of a web browser application, the graphical control objects of the local GUI may include an address

field for entering the address (e.g., URL) of a web page, a refresh button, a go button, and scrollbars. The local GUI is displayed on the local display **560** of the client device **202**. A user at the client device **202** may select graphical control objects of the local GUI displayed on the display **560** using an input device **565**, for example, a touch screen overlying the display **560**. As described further below, the local GUI may be optimized for the display **560** of the client device **202** to provide the user at the client device **202** with a user friendly interface for remotely controlling the application **327** running on the server **204**.

The display module **555** may be configured to receive a local GUI from the local view module **540** and a remote application view of a remote desktop and/or a remote application from the remote view module **535** and generate a composite view comprising the local GUI and the remote application view of the remote desktop and/or remote application.

FIG. **7** shows an example of a composite view rendered on a display **560** of a client device **202**. In this example, the composite view includes a remote application view **704** for viewing an image of the display output of a remote application **327** running on the server **204**. The remote application view is based on the display output data of the remote application **327** received from the server **204**. The user may manipulate the image of the remote application **327** shown in the remote application view **704**. For example, the remote application view **704** may be responsive to, for example, viewing selections by a user such as zooming factor, the depth of the color palette used, viewing angle (pan and tilt) and so on. The viewing selections may be provided in a GUI. The user may also zoom in and out within the remote application view **704** using any technique. For example, the user may zoom in by sliding two fingers on the display **560** away from each other and zoom out by sliding the fingers on the display towards each other on the display **560**, which is currently supported by, for example, iPhone and Palm Pre. User inputs within the remote application view **704** are directed to the remote view module **535**.

The composite view may also include a local GUI **702** and **706**. The local GUI **702** may display a local header for the application shown in the display area **704**. In one aspect, the local header may display an application name based on information received from the server **204**. In another aspect, the local header may be locally generated (e.g., a remote session window). In yet another aspect, the local header may display an editable text input area, such as an address field for a web browser. The local GUI **706** may display, for example, local control GUI objects such as a local control GUI toolbar. The local control GUI objects may be based on information received from the remote server **204**.

The dimensions and layouts of the remote application view **704** and local GUI **702** and **706** are not limited to the example shown in FIG. **7**. The remote application view **704** and local GUI **702** and **706** may have other dimensions and layouts, for example, based on the remote application being accessed by the client device **202**.

In one aspect, the local GUI **702** and **706** is not generated from the display output of a remote application **327** running on the server **204**. For example, the local GUI **702** and **706** is not an image of the remote application's GUI generated from the display output of the remote application **327** running on the server **204**. Instead, the local GUI is locally generated by the local view module **540**, for example, based on configuration file **541** (shown in FIG. **5**B). Furthermore, the configuration file **541** may be received by the local view module **540** from the application control module **325** or the agent module **322**.

FIG. **8**A shows an example of a local control GUI **706**. The local control GUI **706** comprises a toolbar **806** including graphical control objects. The graphical control objects may include a keyboard icon **820**, a BACK arrow **822**, a FORWARD arrow **824**, a pointer icon **826**, a MOUSE icon **828** and a TOOLS icon **830**. When a user selects the keyboard icon **820** (e.g., by touching or tapping), the local view module **540** may change the local control GUI **806** to display a graphical keyboard **820***a* for the user to enter text (shown in FIG. **8**B). The layout details of a keyboard **820***a* may be provided in a configuration file **541** (shown in FIG. **5**B), or may be based on the local GUI tools provided by the OS **460** of the client device **202**.

When a user selects the BACK icon **822**, the web browser application displays a previously displayed web page. In certain embodiments, the client device **202** may perform this by communicating to the server **204**, a control code corresponding to this event. In other embodiments, the client device **202** may include a local cache of previously displayed web pages and may use the cached data to render the previously displayed web page. In a manner similar to the BACK icon **822**, the FORWARD icon **824** may be used to render a later viewed web page in the web browser. The pointer icon **826** may allow a user to activate a pointer within the display area of the remote application view **704**. The MOUSE icon **828** may allow a user to activate a mouse cursor, which then will invoke mouse movements or mouse clicks, and the TOOLS icon **830** may open further menus containing additional tools, as provided in a configuration file and as provided by the OS **460** of the client device **202**.

It will be appreciated by one skilled in the art that, according to one aspect of the disclosure, because the control objects of the local GUI are rendered locally, these objects can be advantageously rendered at the resolution of the local display **560** and can be rendered to have dimensions that can be easily navigated or handled by a user using a finger, a stylus or any other native navigation method(s) provided by the client device **202** (e.g., track-ball, keyboard, etc.). Furthermore, the available display area for the remote application view (e.g., **704**) of a remote application can be fully dedicated to content displayed by the remote application by deactivating control GUI objects from the remote application view **704**. For example, a web browser executed at a remote server **204** may be displayed in the remote application view **704** after deactivating scrollbars and menu header of the web browser at the server **204**. The toolbar and the menu header controls can be provided by the local GUI instead.

FIG. **9** illustrates an example of a composite view **900** that may be rendered locally on display **560**. The composite view **900** comprises the local GUI **902** and **906** and the remote application view **904** of a remote application **327** running on the server **204**. In this example, the graphical control objects of the local GUI **902** and **906** include an IP address field, a LOAD button, a BACK button, a FORWARD button, a MOUSE button, and a GEAR button for remotely controlling a web browser application **327** running on the server **204**.

Referring to the figures discussed above, various operations of the modules in a server **204** and in a client device **202** are further described below in accordance with one aspect of the disclosure.

Now referring to FIGS. **5**B and **7**, the client device **202** may further comprise an input device **560** and a user input

module **550**. In one example, the input device **560** comprises a touch screen overlaying the display **560**. In this aspect, the user may enter user inputs within a display area corresponding to the local GUI **702** and **706** by tapping on a desired graphical control object using a finger or stylus. The user input module **550** may send user inputs to the local GUI **702** and **706** for remotely controlling the application **327** to the local view module **540**. For the example of a touch screen, when the user touches a display area corresponding to the local GUI **702** and **706**, the user input module **550** directs the corresponding user inputs to the local view module **540**. The user inputs may comprise coordinates of the location where the user touched the display **560**.

The user may also enter user inputs within a remote application view **704** of a remote application **327** on the display **560**. For the example of a touch screen, the user may enter user inputs by moving a pointer (not shown) within a display area corresponding to the remote application view **704** of the remote application **327**. In this aspect, the user input module **550** directs user inputs within the remote view **704** to the remote view module **535**.

In one aspect of the disclosure, the local view module **540** may be configured to generate a command message based on user inputs to the local GUI **702** and **706** for remotely controlling the application **327** and send the command message to the server **204**. For the example of a touch screen, the local view module **540** interprets user inputs to the local GUI **702** or **706** and then translates them into corresponding command messages. In this example, a user input may comprise coordinates of a location where the user touches the local GUI **702** or **706** on the display **560** or coordinates of a pointer within the local GUI **702** or **706** controlled by the user, for example, using a pointer device. The local view module **540** can determine which graphical control object is selected by the user based on the graphical control object in the local GUI **702** and **706** corresponding to the coordinates of the user inputs. The local view module **540** determines the function associated with the selected graphical control object and generates a command message for the server **204** with a command to perform the function. For example, if the user touches the local GUI **702** or **706** (e.g., using a finger) at a location corresponding to a refresh button in the local GUI **702** and **706**, then the local view module **540** generates a command message for the server **204** to refresh the web page.

In one aspect, the agent module **322** on the server **204** receives the command message and issues a command to the application control module **325** to control the application **327** based on the command message. For example, if the command message is to refresh the web page, then the agent module **322** instructs the application control module **325** to reload the web page on the web browser application **327**. Thus, the local view module **540** generates command messages based on user inputs to the local GUI and sends the command messages to the agent module **322** on the server **204**, and the agent module **322** controls the remote application based on the received command messages using the application control module **325**.

In one aspect, when a local view module **540** receives one or more user inputs directed to, or placed into, a local GUI (e.g., **702** or **706**), the local view module **540** may interpret the one or more user inputs, determine (or identify) function (s) corresponding to the one or more user inputs, and generate one or more command messages corresponding to the function(s) by, for example, translating the one or more user inputs into the command message(s). Determining the function of a user input may, for example, include identifying the type of user input (e.g., a text entry, a button selection, a menu selection) and/or determining a function corresponding to the user input (e.g., a text entry for an IP address field, a selection of a BACK button, etc.). For example, when the user input comprises a selection of a button (e.g., refresh button) in the local GUI (e.g., based on coordinates of a user's touch on the display **560**), the local view module **540** may determine the function (e.g., refresh current web page) associated with the selected button. In one aspect, these functions are predetermined. The local view module **540** then generates a command message based on the determined function. Some examples of command messages may include, but are not limited to, a command to load a web page from an IP address, display a previously viewed web page, display a later viewed web page, refresh or reload a current web page, stop loading of a webpage, zoom in or out, switch applications, open bookmarks or history (e.g., for a web browser application), and other commands. A command message may comprise a command packet that is transmitted from the client device **202** to the server **204** using various protocols, compressions and encryption schemes. The server **204** may also send commands and status information to the client device **202**. For example, the server **204** may send the client device **202** a web page loading status, a redirected URL or keyboard state.

In one aspect, the remote view module **535** may be configured to send user inputs received within the remote application view **704** to the server **204** via the remote access module **452**. The user inputs may include pointer inputs comprising coordinates of pointer movements and clicks (e.g., mouse clicks). For example, the user may move a pointer (not shown) within the remote application view **704** using a touch screen, a touch pad, a trackball, a mouse or other pointer input device. In this example, the coordinates of the pointer movements may be transmitted to the server **204**. The user inputs may also include keyboard inputs. The user inputs may enter keyboard inputs using a graphical keyboard (e.g., **820***a*) displayed on the display **560**, a keypad or other device. For example, when the user desires to enter text at the location of a pointer or cursor within the remote application view **704**, the user may tap on a keyboard icon (e.g., **920**) to bring up the graphical keyboard to enter the text. In one aspect, the client device access module **452** may send user inputs comprising pointer inputs (e.g., coordinates of pointer movements) and keyboard inputs to the server **204** using RDP, ICA or other remote access protocol.

In one aspect, the server remote access module **334** may receive the user inputs from the client remote access module **452** over the remote access connection **525** and sends the user inputs to the application control module **325**. The application control module **325** interprets the received user inputs and controls the application **327** accordingly.

In one aspect, when an application **327** updates its display output in response to a received command message or user inputs, the remote access module **334** may send updated display output data to the client device **202**. The client remote access module **452** receives the updated display output data of the remote application **327** and sends the updated display output data to the remote view module **535**. The remote view module **535** then generates an updated image of the display output of the remote application, which is displayed within the remote application view **704** of the display **560**.

Aspects of the disclosure allow the user at the client device **202** to remotely view an application running on the

server **204** while controlling the application using a local GUI that can be optimized for the display **560** of the client device **202**.

An advantage of aspects of the disclosure may be illustrated with reference to FIGS. **9** and **1**B. FIG. **9** shows an example of a composite view **900** at the client device **202** for a web browser application running on the server **204**, according to an aspect of the disclosure. In this example, the GUI of the application **327** running on the server **204** may be similar to the GUI **122** shown in FIG. **1**B, which is designed for a large display. Instead of displaying an image of the remote application's GUI **122** running on the server **204**, the local view module **540** advantageously generates and displays a local GUI **902** and **906** for controlling the remote application **327**. In one aspect, the local GUI **902** and **906** in FIG. **9** is not based on the display output of the remote application **327**. The local GUI **902** and **906** may be optimized for the display **560** of the client device **202**. An image of the display output of the remote web browser application **327** is displayed within the remote application view **904** allowing the user at the client device **202** to remotely view the web browser application **327** while remotely controlling the web browser application **327** using the local GUI **902** and **906**.

The local GUI **702** and **706** may be designed to have a similar look and feel of a web browser application that is native to the client device (e.g., smartphone). This allows the user at the client device to control a web browser application **327** running on a server **204** in a similar manner as a web browser that is native to the client device (e.g., smartphone), while enjoying the benefits of server-side computing such as increased computing power and resources available on the server **204**. For example, a web browser application **327** running on a server has access to plug-in applications on the server **204** for rendering multi-media content. As a result, the plug-in applications do not have to be, for example, loaded onto the client device **202**. Another advantage of the local GUI is that it allows a user at the client device **202** to use the application on the server **204** in a user friendly environment without having to extensively rewrite the application port the application to the client device.

In an aspect of the disclosure, the local GUI and the remote application view **704** on the display **560** can be controlled independently. For example, the remote view module **535** may allow the user to adjust the zoom and scroll the image of the remote application **327** within the remote application view **704** while the local view module **540** maintains the size and location of the local GUI **702** and **706** on the display **560**. Thus, the local GUI **702** and **706** can remain on the display **560** and be readily accessible to the user while the user manipulates the image of the remote application within the remote application view **704**.

In an aspect of the disclosure, the agent module **322** may instruct the application control module **325** to deactivate the GUI of the application **327** so that display output data of the application **327** sent to the client device **202** does not include the GUI of the application. For the example of the web browser **120** running on a server in FIG. **1**B, the agent module **322** may deactivate the GUI **122** so that only the display of the web content **124** is sent to the client device **202**. This allows a user at the client device **202** to view the web content **124** of the web browser **120** running on the server **204** while controlling the web browser **120** with the local GUI **702** and **706** instead of the remote application's GUI **122** running on the server **204**.

The local GUI at the client device **202** is not limited to the example of a web browser application and may be used to remotely control many other applications on the server **204**. Examples of applications that may be controlled by local GUIs include word processing applications, spread sheet applications, multi-media player applications, electronic mail applications, and other applications. For the example of a word processing application, the local GUI may include graphical control objects for print, edit, and formatting functions. For a multi-media player, the local GUI may include graphical control objects for play, stop and rewind functions.

In one aspect, the client device **202** may have a plurality of local GUIs for controlling different applications. In this aspect, when the user at the client device **202** remotely launches an application on the server **204**, the agent module **322** may determine what type of application was launched. The agent module **322** may then send a message to the client device **202** via the server remote access module **334** indicating the type of application that was launched on the server **204**. The client remote access module **452** sends the received message to the local view module **540**. The local view module **540** may then select a local GUI from a plurality of local GUIs that matches the type of application indicated by the received message. The agent **322** may also send a configuration file to the client device **202** specifying the local GUI. Thus, the local view module **540** can change the local GUI to adapt to changes in the application running on the server **204**.

FIG. **10** is a conceptual block diagram illustrating a server remote access module **334** and a client remote access module **452** according to an aspect of the disclosure. In one aspect, a server remote access module **334** comprises first and second server remote connection modules **1020***a* and **1020***b* and the client remote access module **452** comprises first and second client remote connection modules **1030***a* and **1030***b*. The first server remote connection module **1020***a* and the first client remote connection module **1030***a* communicate over connection **1025***a*. The second server remote connection module **1020***b* and the second client remote connection module **1030***b* communicate over connection **1025***b*. The connections **1025***a* and **1025***b* may be established between the server **204** and the client device **202** over a network **206** using different sockets. In this aspect, different types of information may be communicated over the different connections **1025***a* and **1025***b*.

In one aspect of the disclosure, the first server remote connection module **1020***a* and the first client remote connection module **1030***a* are implemented using a remote access application (e.g., RDP application) for communicating display output data and user inputs between the server **204** and the client device **202**. The connection **1025***a* may comprise a plurality of virtual channels with different virtual channels carrying display output data (e.g., images or audio output of a remote application) and user inputs (e.g., pointer and keyboard inputs from a user at a client device) between the server **204** and the client device **202**.

The second server remote connection module **1020***b* and the second client remote connection module **1030***b* may be configured to communicate command messages over the connection **1025***b*. In this aspect, the second client remote connection module **1030***b* sends command messages from the local view module **540** to the server **204** over the connection **1025***b*. The second client remote connection module **1020***b* communicates the received command messages to the agent module **322**. In this aspect, the agent module **322** may manage the connection **1025***b*. In one aspect, the second remote connection module **1020***b* may be integrated with the agent module **322**. Thus, in this aspect,

the command messages are communicated over a different connection than the display data and user inputs. An advantage of this aspect is that the command messages may be communicated using a different protocol than the display data and user inputs. For example, the display output data and user inputs may be communicated using a remote access protocol (e.g., RDP) while the command messages are communicated using a different protocol. For example, the command messages may be communicated using a protocol having different encryption, compression and/or security features than the remote access protocol.

FIG. **11** depicts an example of the connection **1025***a* comprising multiple virtual channels **1120***a* to **1120***c* and the connection **1025***b*. In this example, the connection **1025***a* may be based on a remote access protocol (e.g., RDP) for communicating display output data and user inputs on different virtual channels **1120***a* to **1120***c*. The connection **1025***b* is used to communicate command messages. Alternatively, the command messages may be sent over the connection **1025***a*, in which case the second remote connection modules **1020***b* and **1030***b* may be omitted. For example, the command messages may be sent on a virtual channel of the connection **1025***a* established for command messages. In an aspect, a command message may include a flag or other indicator identifying the message as a command message.

FIGS. **10** and **11** illustrate merely examples of communications between a server and a client device. A server may include one or more remote connection modules (e.g., one, two, three or more), and a client device include one or more remote connection modules (e.g., one, two, three or more).

FIG. **12** is a conceptual block diagram illustrating an example of a system.

A system **1200** may be, for example, a client device or a server. The system **1200** includes a processing system **1202**. The processing system **1202** is capable of communication with a receiver **1206** and a transmitter **1209** through a bus **1204** or other structures or devices. It should be understood that communication means other than busses can be utilized with the disclosed configurations. The processing system **1202** can generate audio, video, multimedia, and/or other types of data to be provided to the transmitter **1209** for communication. In addition, audio, video, multimedia, and/or other types of data can be received at the receiver **1206**, and processed by the processing system **1202**.

The processing system **1202** may include a general-purpose processor or a specific-purpose processor for executing instructions and may further include a machine-readable medium **1219**, such as a volatile or non-volatile memory, for storing data and/or instructions for software programs. The instructions, which may be stored in a machine-readable medium **1210** and/or **1219**, may be executed by the processing system **1202** to control and manage access to the various networks, as well as provide other communication and processing functions. The instructions may also include instructions executed by the processing system **1202** for various user interface devices, such as a display **1212** and a keypad **1214**. The processing system **1202** may include an input port **1222** and an output port **1224**. Each of the input port **1222** and the output port **1224** may include one or more ports. The input port **1222** and the output port **1224** may be the same port (e.g., a bi-directional port) or may be different ports.

The processing system **1202** may be implemented using software, hardware, or a combination of both. By way of example, the processing system **102** may be implemented with one or more processors. A processor may be a general-purpose microprocessor, a microcontroller, a Digital Signal Processor (DSP), an Application Specific Integrated Circuit (ASIC), a Field Programmable Gate Array (FPGA), a Programmable Logic Device (PLD), a controller, a state machine, gated logic, discrete hardware components, or any other suitable device that can perform calculations or other manipulations of information.

A machine-readable medium can be one or more machine-readable media. Software shall be construed broadly to mean instructions, data, or any combination thereof, whether referred to as software, firmware, middleware, microcode, hardware description language, or otherwise. Instructions may include code (e.g., in source code format, binary code format, executable code format, or any other suitable format of code).

Machine-readable media (e.g., **1219**) may include storage integrated into a processing system, such as might be the case with an ASIC. Machine-readable media (e.g., **1210**) may also include storage external to a processing system, such as a Random Access Memory (RAM), a flash memory, a Read Only Memory (ROM), a Programmable Read-Only Memory (PROM), an Erasable PROM (EPROM), registers, a hard disk, a removable disk, a CD-ROM, a DVD, or any other suitable storage device. In addition, machine-readable media may include a transmission line or a carrier wave that encodes a data signal. Those skilled in the art will recognize how best to implement the described functionality for the processing system **1202**. According to one aspect of the disclosure, a machine-readable medium is a computer-readable medium encoded or stored with instructions and is a computing element, which defines structural and functional interrelationships between the instructions and the rest of the system, which permit the instructions' functionality to be realized. Instructions may be executable, for example, by a client device or server or by a processing system of a client device or server. Instructions can be, for example, a computer program including code.

An interface **1216** may be any type of interface and may reside between any of the components shown in FIG. **12**. An interface **1216** may also be, for example, an interface to the outside world (e.g., an Internet network interface). A transceiver block **1207** may represent one or more transceivers, and each transceiver may include a receiver **1206** and a transmitter **1209**. A functionality implemented in a processing system **1202** may be implemented in a portion of a receiver **1206**, a portion of a transmitter **1209**, a portion of a machine-readable medium **1210**, a portion of a display **1212**, a portion of a keypad **1214**, or a portion of an interface **1216**, and vice versa.

The subject technology is illustrated, for example, according to various aspects described below. Numbered clauses are provided below for convenience. These are provided as examples, and do not limit the subject technology.

1. A system for rendering at a local client side a composite view including a local graphical user interface and a remote application view associated with a remote application running at a remote server, comprising:

a local view module configured to generate a local graphical user interface (GUI) for controlling the remote application remotely from the local client side, configured to receive one or more user inputs to the local GUI, configured to determine a function corresponding to the one or more user inputs, configured to generate a command message based on the determined function, and configured to direct the command message to the remote server;

a remote view module configured to generate a remote application view of the remote application based on display

output data of the remote application received at the local client side from the remote server; and

a display module configured to provide, to a local display at the local client side, a composite view comprising the local GUI and the remote application view of the remote application.

2. The system of clause 1, wherein the local view module is configured to receive a message identifying a type of application running on the remote server and to select the local GUI from a plurality of local GUIs based on the received message.

3. The system of clause 1, wherein the local view module is configured to be controlled dynamically by the remote server to generate the local GUI based on the application running on the remote server.

4. The system of clause 1, wherein the remote view module is configured to direct user inputs within the remote application view to the remote server.

5. The system of clause 4, wherein the user inputs from the remote view module comprise pointer inputs, keyboard inputs or finger inputs (e.g., gestures).

6. The system of clause 1, wherein the display output data includes an image or drawing commands of a display output of the remote application.

7. The system of clause 1, wherein the one or more user inputs to the local GUI include coordinates of a pointer or a user's touch on the local display, and the local view module is configured to determine the function based on a graphical control object of the local GUI corresponding to the coordinates of the pointer or the user's touch on the local display.

8. The system of clause 1, wherein the local view module is configured to control a size and a location of the local GUI on the local display independently from the remote application view of the remote application.

9. The system of clause 1, wherein the local view module is configured to direct the command message to the remote server through a remote access module at the local client side.

10. The system of clause 9, wherein the remote access module is configured to communicate with the remote server over a first connection and a second connection via different socket connections, to receive the display output data of the remote application and to send user inputs over the first connection and to send the command message over the second connection.

11. The system of clause 9, wherein the remote access module is configured to communicate with the remote server over a plurality of virtual channels, to send user inputs over a first one of the plurality of virtual channels and to send the command message over a second one of the plurality of virtual channels.

12. A method for rendering at a local client side a composite view including a local graphical user interface and a remote application view associated with a remote application running at a remote server, comprising:

generating a local graphical user interface (GUI) for remotely controlling the remote application from the local client device (e.g., **1310**-A of FIG. **13**A);

generating a remote view of the remote application based on display output data of the remote application received at the local client device from the remote server (e.g., **1320**-A);

providing, to a local display, a composite view comprising the local GUI and the remote view of the remote application (e.g., **1330**-A);

determining a function based on one or more user inputs to the local GUI (e.g., **1340**-A);

generating a command message based on the determined function (e.g., **1350**-A); and

directing the command message to the remote server (e.g., **1360**-A).

13. The method of clause 12, further comprising:

receiving a message identifying a type of application running on the remote server; and

selecting the local GUI from a plurality of local GUIs based on the received message.

14. The method of clause 12, further comprising directing user inputs within the remote application view to the remote server.

15. The method of clause 14, wherein the user inputs from the remote view module comprise pointer inputs or keyboard inputs.

16. The method of clause 12, wherein the display output data includes an image or drawing commands of a display output of the remote application.

17. The method of clause 12, wherein the one or more user inputs to the local GUI include coordinates of a pointer or a user's touch on the local display, and the determining the function based on the one or more user inputs comprises:

determining the function based on a graphical control object of the local GUI corresponding to the coordinates of the pointer or the user's touch on the local display.

18. The method of clause 12, further comprising controlling a size and a location of the local GUI on the local display independently from the remote application view of the remote application.

19. The method of clause 12, wherein the directing the command message to the remote server comprises directing the command message to the remote server through a remote access module at the local client side.

20. The method of clause 19, further comprising:

communicating with the remote server over a first connection and a second connection via different socket connections; and

sending user inputs over the first connection,

wherein the receiving the display output data comprises receiving the display output data of the remote application over the first connection, and the directing the command message to the remote server comprises sending the command message over the second connection.

21. The method of clause 19, further comprising:

communicating with the remote server over a plurality of virtual channels; and

sending user inputs over a first one of the plurality of virtual channels,

wherein the directing the command message to the remote server comprises sending the command message over a second one of the plurality of virtual channels.

22. A machine-readable medium encoded with instructions for rendering at a local client side a composite view including a local graphical user interface and a remote application view associated with a remote application running at a remote server, the instructions comprising code for:

generating a local graphical user interface (GUI) for remotely controlling the remote application from the local client device;

generating a remote view of the remote application based on display output data of the remote application received at the local client device from the remote server;

providing, to a local display, a composite view comprising the local GUI and the remote view of the remote application;

determining a function based on one or more user inputs to the local GUI;

generating a command message based on the determined function; and

directing the command message to the remote server.

23. The machine-readable medium of clause 22, wherein the instructions further comprise code for:

receiving a message identifying a type of application running on the remote server; and

selecting the local GUI from a plurality of local GUIs based on the received message.

24. The machine-readable medium of clause 22, wherein the instructions further comprise code for directing user inputs within the remote application view to the remote server.

25. The machine-readable medium of clause 24, wherein the user inputs from the remote view module comprise pointer inputs, keyboard inputs, or finger inputs (e.g., touch inputs).

26. The machine-readable medium of clause 22, wherein the display output data includes an image or drawing commands of a display output of the remote application.

27. The machine-readable medium of clause 22, wherein the one or more user inputs to the local GUI include coordinates of a pointer or a user's touch on the local display, and the determining the function based on the one or more user inputs comprises:

determining the function based on a graphical control object of the local GUI corresponding to the coordinates of the pointer or the user's touch on the local display.

28. The machine-readable medium of clause 22, wherein the instructions further comprise code for controlling a size and a location of the local GUI on the local display independently from the remote application view of the remote application.

29. The machine-readable medium of clause 22, wherein directing the command message to the remote server comprises directing the command message to the remote server through a remote access module at the local client side.

30. The machine-readable medium of clause 29, wherein the instructions further comprise code for:

communicating with the remote server over a first connection and a second connection via different socket connections; and

sending user inputs over the first connection,

wherein the receiving the display output data comprises receiving the display output data of the remote application over the first connection, and the directing the command message to the remote server comprises sending the command message over the second connection.

31. The machine-readable medium of clause 29, wherein the instructions further comprise code for:

communicating with the remote server over a plurality of virtual channels; and

sending user inputs over a first one of the plurality of virtual channels,

wherein the directing the command message to the remote server comprises sending the command message over a second one of the plurality of virtual channels.

32. An apparatus for rendering at a local client side a composite view including a local graphical user interface and a remote application view associated with a remote application running at a remote server, comprising:

means for generating a local graphical user interface (GUI) for remotely controlling the remote application from the local client device (e.g., **1310**-B of FIG. **13**B);

means for generating a remote view of the remote application based on display output data of the remote application received at the local client device from the remote server (e.g., **1320**-B);

means for providing, to a local display, a composite view comprising the local GUI and the remote view of the remote application (e.g., **1330**-B);

means for determining a function based on one or more user inputs to the local GUI (e.g., **1340**-B);

means for generating a command message based on the determined function (e.g., **1350**-B); and

means for directing the command message to the remote server (e.g., **1360**-B).

33. The apparatus of clause 32, further comprising:

means for receiving a message identifying a type of application running on the remote server; and

means for selecting the local GUI from a plurality of local GUIs based on the received message.

34. The apparatus of clause 32, further comprising means for directing user inputs within the remote application view to the remote server.

35. The apparatus of clause 34, wherein the user inputs from the remote view module comprise pointer inputs or keyboard inputs.

36. The apparatus of clause 32, wherein the display output data includes an image or drawing commands of a display output of the remote application.

37. The apparatus of clause 32, wherein the one or more user inputs to the local GUI include coordinates of a pointer or a user's touch on the local display, and the means for determining the function based on the one or more user inputs comprising:

means for determining the function based on a graphical control object of the local GUI corresponding to the coordinates of the pointer or the user's touch on the local display.

38. The apparatus of clause 32, further comprising means for controlling a size and a location of the local GUI on the local display independently from the remote application view of the remote application.

39. The apparatus of clause 32, wherein the means for directing the command message to the remote server comprises means for directing the command message to the remote server through a remote access module at the local client side.

40. The apparatus of clause 39, further comprising:

means for communicating with the remote server over a first connection and a second connection via different socket connections; and

means for sending user inputs over the first connection,

wherein the means for receiving the display output data comprises means for receiving the display output data of the remote application over the first connection, and the means for directing the command message to the remote server comprises means for sending the command message over the second connection.

41. The apparatus of clause 39, further comprising:

means for communicating with the remote server over a plurality of virtual channels; and

means for sending user inputs over a first one of the plurality of virtual channels,

wherein the means for directing the command message to the remote server comprises means for sending the command message over a second one of the plurality of virtual channels.

Those of skill in the art would appreciate that the various illustrative blocks, modules, elements, components, meth-

ods, and algorithms described herein may be implemented as electronic hardware, computer software, or combinations of both.

For example, a module (e.g., an agent module **322**, a viewer controller module **456**, a local view module **540**, a remote view module **535**, or any other modules) may be implemented as electronic hardware, computer software, or combinations of both. In one aspect, a module(s) may be an apparatus since a module(s) may include instructions encoded or stored on a machine-readable medium, on another device, or on a portion thereof. In one aspect, a module(s) may be software (e.g., an application, a subroutine) stored in a machine-readable medium and executable by a processing system or a processor. In another aspect, a module(s) may be hardware (e.g., machine-readable medium encoded with instructions, a pre-programmed general-purpose computer, or a special purpose electronic or optical device).

Various modules may reside in one machine or in multiple machines. In one example, modules for the server side (e.g., an agent module, an application module, a server remote access module, etc.) may be located in one server or spread over multiple servers. In another example, modules for the client side (e.g., a client remote access module, a viewer controller module, a local view module, a remote view module, a display module, etc.) may be located in one client device or spread over multiple client devices.

In one aspect of the disclosure, when actions or functions are described as being performed by a module or a component (e.g., establishing, sending, receiving, providing, building, displaying, registering, encrypting, decrypting, authenticating, notifying, accepting, selecting, controlling, issuing, transmitting, reporting, pushing, or any other action or function), it is understood that such actions or functions are performed by the module or the component directly or indirectly. As an example, when a module is described as performing an action, it is understood that the module may perform the action directly or may perform the action indirectly, for example, by facilitating such an action. For instance, when a session is described as being established by a module, it is understood that the module may establish the session indirectly by facilitating an establishment of the session. As yet another example, when a view of an application is described as being displayed or rendered by a module, it is understood that the view may be displayed or rendered by the module either directly or indirectly.

To illustrate this interchangeability of hardware and software, various illustrative blocks, modules, elements, components, methods, and algorithms have been described above generally in terms of their functionality. Whether such functionality is implemented as hardware or software depends upon the particular application and design constraints imposed on the overall system. Skilled artisans may implement the described functionality in varying ways for each particular application.

Various components and blocks may be arranged differently (e.g., arranged in a different order, or partitioned in a different way) all without departing from the scope of the subject technology. In one aspect of the disclosure, the modules (or elements) recited in the accompanying claims may be performed by one module or by a smaller number of modules, and this arrangement is within the scope of the claims. In another aspect, the modules (or elements) recited in the accompanying claims may be performed by a larger number of modules, and this arrangement is within the scope of the claims. In yet another aspect, a module (or an element) recited in the accompanying claims may be performed by multiple modules, and this arrangement is within the scope of the claims. For example, a local view module and a remote view module may be combined into one module. A client remote access module, a local view module and a remote view module may be combined into one module. An agent module and a server remote access module may be combined into one module. In another example, these modules may be divided into a larger number of modules.

It is understood that the specific order or hierarchy of steps in the processes disclosed is an illustration of exemplary approaches. Based upon design preferences, it is understood that the specific order or hierarchy of steps in the processes may be rearranged. Some of the steps may be performed simultaneously. The accompanying method claims present elements of the various steps in a sample order, and are not meant to be limited to the specific order or hierarchy presented.

The previous description is provided to enable any person skilled in the art to practice the various aspects described herein. The previous description provides various examples of the subject technology, and the subject technology is not limited to these examples. Various modifications to these aspects will be readily apparent to those skilled in the art, and the generic principles defined herein may be applied to other aspects. Thus, the claims are not intended to be limited to the aspects shown herein, but is to be accorded the full scope consistent with the language claims, wherein reference to an element in the singular is not intended to mean "one and only one" unless specifically so stated, but rather "one or more." Unless specifically stated otherwise, the term "some" refers to one or more. Pronouns in the masculine (e.g., his) include the feminine and neuter gender (e.g., her and its) and vice versa. Headings and subheadings, if any, are used for convenience only and do not limit the invention.

A phrase such as an "aspect" does not imply that such aspect is essential to the subject technology or that such aspect applies to all configurations of the subject technology. A disclosure relating to an aspect may apply to all configurations, or one or more configurations. An aspect may provide one or more examples of the disclosure. A phrase such as an aspect may refer to one or more aspects and vice versa. A phrase such as an "embodiment" does not imply that such embodiment is essential to the subject technology or that such embodiment applies to all configurations of the subject technology. A disclosure relating to an embodiment may apply to all embodiments, or one or more embodiments. An embodiment may provide one or more examples of the disclosure. A phrase such an embodiment may refer to one or more embodiments and vice versa. A phrase such as a "configuration" does not imply that such configuration is essential to the subject technology or that such configuration applies to all configurations of the subject technology. A disclosure relating to a configuration may apply to all configurations, or one or more configurations. A configuration may provide one or more examples of the disclosure. A phrase such a configuration may refer to one or more configurations and vice versa.

The word "exemplary" is used herein to mean "serving as an example or illustration." Any aspect or design described herein as "exemplary" is not necessarily to be construed as preferred or advantageous over other aspects or designs.

All structural and functional equivalents to the elements of the various aspects described throughout this disclosure that are known or later come to be known to those of ordinary skill in the art are expressly incorporated herein by reference and are intended to be encompassed by the claims. Moreover, nothing disclosed herein is intended to be dedi-

cated to the public regardless of whether such disclosure is explicitly recited in the claims. No claim element is to be construed under the provisions of 35 U.S.C. §112, sixth paragraph, unless the element is expressly recited using the phrase "means for" or, in the case of a method claim, the element is recited using the phrase "step for." Furthermore, to the extent that the term "include," "have," or the like is used in the description or the claims, such term is intended to be inclusive in a manner similar to the term "comprise" as "comprise" is interpreted when employed as a transitional word in a claim.

What is claimed is:

1. A system for rendering at a local client side a composite view including a local graphical user interface and a remote application view associated with a remote application of a remote server, comprising:

a network interface;

a processor; and

a memory communicatively coupled to the processor, the memory comprising instructions operable, when executed by the processor, to cause the processor to:

generate a local graphical user interface (GUI) for controlling the remote application remotely from the local client side, the local client side is a mobile device,

receive one or more user inputs to the local GUI,

determine a function to be performed by the remote application of the remote server based on the one or more user inputs to the local GUI,

generate a command message identifying the determined function, direct the command message to the remote server via the network interface;

generate a remote application view of the remote application based on display output data of the remote application received at the local client side from the remote server via the network interface, where the remote application view only includes the display output data, the display output data including an image of the display output of the remote application and is used by the local client side to create a composite view, where a remote GUI of the remote application is deactivated in the display output data; and

provide, to a local display at the local client side, the composite view comprising the local GUI and the remote application view of the remote application.

2. The system of claim **1**, wherein the instructions further cause the processor to receive, via the network interface, a message identifying a type of application of the remote server and to select the local GUI from a plurality of local GUIs based on the received message.

3. The system of claim **1**, wherein the instructions further cause the processor to be controlled dynamically by the remote server to generate the local GUI based on the application of the remote server.

4. The system of claim **1**, wherein the instructions further cause the processor to direct user inputs within the remote application view to the remote server.

5. The system of claim **4**, wherein the user inputs comprise pointer inputs, keyboard inputs or finger inputs.

6. The system of claim **1**, wherein the display output data includes an image or drawing commands of a display output of the remote application.

7. The system of claim **1**, wherein the one or more user inputs to the local GUI include coordinates of a pointer or a user's touch on the local display, and the instructions further cause the processor to determine the function based on a graphical control object of the local GUI corresponding to the coordinates of the pointer or the user's touch on the local display.

8. The system of claim **1**, wherein the instructions further cause the processor to control a size and a location of the local GUI on the local display independently from the remote application view of the remote application.

9. The system of claim **1**, wherein the instructions further cause the processor to direct the command message to the remote server through a remote access module at the local client side.

10. The system of claim **9**, wherein the instructions further cause the processor to communicate with the remote server over a first connection and a second connection via different socket connections, to receive the display output data of the remote application and to send user inputs over the first connection and to send the command message over the second connection.

11. The system of claim **9**, wherein the instructions further cause the processor to communicate with the remote server over a plurality of virtual channels, to send user inputs over a first one of the plurality of virtual channels and to send the command message over a second one of the plurality of virtual channels.

12. A method for rendering at a local client side a composite view including a local graphical user interface and a remote application view associated with a remote application of a remote server, comprising:

generating a local graphical user interface (GUI) for remotely controlling the remote application from the local client device, the local client device is a mobile device;

generating a remote view of the remote application based on display output data of the remote application received at the local client device from the remote server, where the remote application view only includes the display output data, the display output data including an image of the display output of the remote application and is used by the local client device to create a composite view, where a remote GUI of the remote application is deactivated in the display output data;

providing, to a local display, the composite view comprising the local GUI and the remote view of the remote application;

determining a function to be performed by the remote application of the remote server based on one or more user inputs to the local GUI;

generating a command message identifying the determined function; and directing the command message to the remote server.

13. A non-transitory machine-readable medium encoded with instructions for rendering at a local client side a composite view including a local graphical user interface and a remote application view associated with a remote application of a remote server, the instructions comprising code for:

generating a local graphical user interface (GUI) for remotely controlling the remote application from the local client device, the local client device is a mobile device;

generating a remote view of the remote application based on display output data of the remote application received at the local client device from the remote server, where the remote application view only includes the display output data, the display output data including an image of the display output of the remote

application and is used by the local client device to create a composite view, where a remote GUI of the remote application is deactivated in the display output data;

providing, to a local display, the composite view comprising the local GUI and the remote view of the remote application;

determining a function to be performed by the remote application of the remote server based on one or more user inputs to the local GUI;

generating a command message identifying the determined function; and

directing the command message to the remote server.

14. The non-transitory machine-readable medium of claim **13**, wherein the instructions further comprise code for:

receiving, via a network interface, a message identifying a type of application of the remote server; and

selecting the local GUI from a plurality of local GUIs based on the received message.

15. The non-transitory machine-readable medium of claim **13**, wherein the instructions further comprise code for directing user inputs within the remote application view to the remote server.

16. The non-transitory machine-readable medium of claim **15**, wherein the user inputs from the remote application view comprise pointer inputs, keyboard inputs or finger inputs.

17. The non-transitory machine-readable medium of claim **13**, wherein the display output data includes an image or drawing commands of a display output of the remote application.

18. The non-transitory machine-readable medium of claim **13**, wherein the one or more user inputs to the local GUI include coordinates of a pointer or a user's touch on the local display, and determining the function based on the one or more user inputs comprises:

determining the function based on a graphical control object of the local GUI

corresponding to the coordinates of the pointer or the user's touch on the local display.

19. The non-transitory machine-readable medium of claim **13**, wherein the instructions further comprise code for controlling a size and a location of the local GUI on the display independently from the remote application view of the remote application.

20. An apparatus for rendering at a local client side a composite view including a local graphical user interface and a remote application view associated with a remote application of a remote server, comprising:

means for generating a local graphical user interface (GUI) for remotely controlling the remote application from the local client device, the local client device is a mobile device;

means for generating a remote view of the remote application based on display output data of the remote application received at the local client device from the remote server, where the remote application view only includes the display output data, the display output data including an image of the display output of the remote application and is used by the local client device to create a composite view, where a remote GUI of the remote application is deactivated in the display output data;

means for providing, to a local display, the composite view comprising the local GUI and the remote view of the remote application;

means for determining a function to be performed by the remote application of the remote server based on one or more user inputs to the local GUI;

means for generating a command message identifying the determined function; and

means for directing the command message to the remote server.

21. The system of claim **1**, wherein the remote application of the remote server comprises a browser application of the remote server, and the determined function comprises one of a page reload function, a page forward function, and a page backwards function.

* * * * *